Avaya Aura™ SBC Objects and Properties Reference

Notice

While reasonable efforts have been made to ensure that the information in this document is complete and accurate at the time of printing, Avaya assumes no liability for any errors. Avaya reserves the right to make changes and corrections to the information in this document without the obligation to notify any person or organization of such changes.

Documentation disclaimer

Avaya shall not be responsible for any modifications, additions, or deletions to the original published version of this documentation unless such modifications, additions, or deletions were performed by Avaya.

End User agree to indemnify and hold harmless Avaya, Avaya's agents, servants and employees against all claims, lawsuits, demands and judgments arising out of, or in connection with, subsequent modifications, additions or deletions to this documentation, to the extent made by End User.

Link disclaimer

Avaya is not responsible for the contents or reliability of any linked Websites referenced within this site or documentation(s) provided by Avaya. Avaya is not responsible for the accuracy of any information, statement or content provided on these sites and does not necessarily endorse the products, services, or information described or offered within them. Avaya does not guarantee that these links will work all the time and has no control over the availability of the linked pages.

Warranty

Avaya provides a limited warranty on this product. Refer to your sales agreement to establish the terms of the limited warranty. In addition, Avaya's standard warranty language, as well as information regarding support for this product, while under warranty, is available to Avaya customers and other parties through the Avaya Support Website: http://www.avaya.com/support. Please note that if you acquired the product from an authorized Avaya reseller outside of the United States and Canada, the warranty is provided to you by the said Avaya reseller and not by Avaya.

Licenses

The software license terms available on the Avaya Website, http://support.avaya.com/licenseinfo/ are applicable to anyone who downloads, uses and/or installs Avaya software, purchased from Avaya Inc., any Avaya affiliate, or an authorized Avaya reseller (as applicable) under a commercial agreement with Avaya or an authorized Avaya reseller. Unless otherwise agreed to by Avaya in writing, Avaya does not extend this license if the software was obtained from anyone other than Avaya, an Avaya affiliate or an Avaya authorized reseller, and Avaya reserves the right to take legal action against you and anyone else using or selling the software without a license. By installing, downloading or using the software, or authorizing others to do so, you, on behalf of yourself and the entity for whom you are installing, downloading or using the software (hereinafter referred to interchangeably as “you” and “end user”), agree to these terms and conditions and create a binding contract between you and Avaya Inc. Or the applicable Avaya affiliate (“Avaya”).

Avaya grants End User a license within the scope of the license types described below. The applicable number of licenses and units of capacity for which the license is granted will be one (1), unless a different number of licenses or units of capacity is specified in the Documentation or other materials available to End User. “Designated Processor” means a single stand-alone computing device. “Server” means a Designated Processor that hosts a software application to be accessed by multiple users. “Software” means the computer programs in object code, originally licensed by Avaya and ultimately utilized by End User, whether as stand-alone products or pre-installed on Hardware. “Hardware” means the standard hardware originally sold by Avaya and ultimately utilized by End User.

License types

- Designated System(s) License (DS):
  End User may install and use each copy of the Software on only one Designated Processor, unless a different number of Designated Processors is indicated in the Documentation or other materials available to End User. Avaya may require the Designated Processor(s) to be identified by type, serial number, feature key, location or other specific designation, or to be provided by End User to Avaya through electronic means established by Avaya specifically for this purpose.
- Concurrent User License (CU):
  End User may install and use the Software on multiple Designated Processors or one or more Servers, so long as only the licensed number of Units are accessing and using the Software at any given time. A “Unit” means the unit on which Avaya, at its sole discretion, bases the pricing of its licenses and can be, without limitation, an agent, port or user, an e-mail or voice mail account in the name of a person or corporate function (e.g., webmaster or helpdesk), or a directory entry in the administrative database utilized by the Software that permits one user to interface with the Software. Units may be linked to a specific, identified Server.
- Named User License (NU):
  End User may: (i) install and use the Software on a single Designated Processor or Server per authorized Named User (defined below); or (ii) install and use the Software on a Server so long as only authorized Named Users access and use the Software. “Named User” means a user or device that has been expressly authorized by Avaya to access and use the Software. At Avaya's sole discretion, a “Named User” may be, without limitation, designated by name, corporate function (for example, webmaster or helpdesk), an e-mail or voice mail account in the name of a person or corporate function, or a directory entry in the administrative database utilized by the Software that permits one user to interface with the Software.

- Shrinkwrap License (SR):
  Customer may install and use the Software in accordance with the terms and conditions of the applicable license agreements, such as “shrinkwrap” or “clickthrough” license accompanying or applicable to the Software (“Shrinkwrap License”). (See Third-party Components for more information).

Copyright

Third Party Components

Certain software programs or portions thereof included in the Product may contain software distributed under third party agreements (“Third Party Components”), which may contain terms that expand or limit rights to use certain portions of the Product (“Third Party Terms”). Information regarding distributed Linux OS source code (for those Products that have distributed the Linux OS source code), and identifying the copyright holders of the Third Party Components and the Third Party Terms that apply to them is available on the Avaya Support Website: http://support.avaya.com/Copyright.

Preventing toll fraud

“Toll fraud” is the unauthorized use of your telecommunications system by an unauthorized party (for example, a person who is not a corporate employee, agent, subcontractor, or is not working on your company's behalf). Be aware that there can be a risk of toll fraud associated with your system and that, if toll fraud occurs, it can result in substantial additional charges for your telecommunications services.

Avaya fraud intervention

If you suspect that you are being victimized by toll fraud and you need technical assistance or support, call Technical Service Center Toll Fraud Intervention Hotline at +1-800-643-2353 for the United States and Canada. For additional support telephone numbers, see the Avaya Support Website:

http://www.support.avaya.com/.

Suspected security vulnerabilities with Avaya products should be reported to Avaya by sending mail to:securityalerts@avaya.com.

Trademarks

Avaya® and Avaya Aura™ are trademarks of Avaya Inc.

The trademarks, logos and service marks (“Marks”) displayed in this site, the documentation(s) and product(s) provided by Avaya are the registered or unregistered Marks of Avaya, its affiliates, or other third parties. Users are not permitted to use such Marks without prior written consent from Avaya or such third party which may own the Mark. Nothing contained in this site, the documentation(s) and product(s) should be construed as granting, by implication, estoppel, or otherwise, any license or right in and to the Marks without the express written permission of Avaya or the applicable third party.

All non-Avaya trademarks are the property of their respective owners.

Downloading documents

For the most current versions of documentation, see the Avaya Support Website: http://www.avaya.com/support.

Contact Avaya Support

Avaya provides a telephone number for you to use to report problems or to ask questions about your product. The support telephone number is 1-800-242-2121 in the United States. For additional support telephone numbers, see the Avaya Website: http://www.avaya.com/support.

Contents

1. Using the Net-Net 2600 Command Line Interface

About this chapter

This chapter describes using the command line interface (CLI) to manage and monitor your AA-SBC system. For a detailed description of the CLI and for information on selecting a management tool or using the AA-SBC Management System, see the *Net-Net OS-E – Using the NNOS-E Management Tools* guide.

Selecting a management tool

You can configure, manage, and/or monitor AA-SBC using any one of the following interfaces:

- Command line interface (CLI)
- AA-SBC Web-based management system
- SNMP (Simple Network Management Protocol)
- XML (Extensible Markup Language)
- WSDL (Web Services Description Language)

See the *Net-Net OS-E – Using the NNOS-E Management Tools* guide for a complete description of each.

Getting started with the (CLI)

The CLI is a text-based interface that allows you to manage and configure all aspects of AA-SBC. Use the CLI to create, edit, and display the AA-SBC configuration file. In addition, you can display various components of system status and data. You can access the CLI using a PC or system console, Telnet application, or Secure Shell protocols (SSH1 and SSH2).

CLI quick start

The CLI features a quick-start script that allows you to complete basic configuration through prompting.

Accessing the CLI

You access the CLI from a system console (serial connection), Telnet, or SSH.

Using a system console

The following table lists the steps for accessing the CLI using a system console attached to AA-SBC:

Table 0-1.

Step	Action
1.	Connect a PC to the AA-SBC device by connecting an EIA-232 (RS-232) straight-through serial cable between the console port on the system and the PC COM1 or serial port.
2.	Start a terminal emulator on the PC. Tera Term Pro and HyperTerminal are popular terminal emulation programs.
3.	Configure the terminal emulator for: • VT100 emulation, or let it autodetect the terminal type. • 115200 baud, 8 data bits, no parity bit, 1 stop bit (8/N/1), no flow control.
4.	Press [ENTER] on the PC keyboard until the CLI prompt appears: NNOS-E> _

Using Telnet

The following table lists the steps for accessing the CLI using Telnet, the TCP/IP terminal emulation protocol:

Step	Action
1.	Connect the AA-SBC device to a network that the Telnet-client system can reach by connecting a network cable from the Ethernet 0 port on the device to a network patch panel, Ethernet switch, or device.
2.	Configure an IP address for the Ethernet port according to instructions in the *Net-Net OS-E – System Administration Guide* .
3.	Start the Telnet client. The default Telnet port is 23 (although you may configure the AA-SBC and client for a different port number). At the Telnet prompt, enter **open** *ipaddress*, where *ipaddress* is the IP address of the AA-SBC Ethernet port.

4.	Log in using the user name and password that the system administrator assigned to you.
	Example: `username: user1`
	`password: mypassword`
	`NNOS-E>`

Using SSH1 or SSH2 (Secure Shell)

The following table describes how to access the CLI using SSH1 or SSH2 (Secure Shell, a secure Telnet-like terminal emulation protocol). See Supported SSH clients on page 46 for a list of supported SSH clients.

Step	Action
1.	Connect the system to a network that the SSH client system can reach by connecting a UTP/STP Category 5 network cable from the Management port to a network patch panel or device.
2.	Configure an IP address for the Management port according to the instructions in the *Net-Net OS-E – System Administration Guide*.

Step	Action
3.	Start the SSH client and specify the host name or address for the system. If using the password authentication method, the system prompts you for a password. Otherwise, if using a valid public key or no authentication is required, you connect to the system.

4.	Log in using the user name and password that the system administrator assigned to you:
	Example: `username: ` **`secureuser1`**
	`password: ` **`mypassword`**
	`NNOS-E>`

Supported SSH clients

AA-SBC supports the following SSH clients:

- SecureCRT®, for Microsoft Windows platforms.
- PuTTY, for Microsoft Windows platforms.
- OpenSSH, for UNIX and Linux platforms.

CLI basic concepts

The AA-SBC CLI structure consists of a command hierarchy of configurable objects and properties. When you use the CLI to create, edit, or modify the configuration, the software writes the new configuration to the default file named **cxc.cfg**.

It is important to understand the states of the AA-SBC configuration file—saved, running, and working:

1. the working config, which keeps a record of configuration edits
2. the running config, which is used by the system
3. the saved config, which the system boots from.

Before using the CLI, refer to the *Net-Net OS-E – Using the NNOS-E Management Tools* guide for a description of each state.

CLI structure summary

The AA-SBC configuration file is a hierarchy of objects and properties; objects and properties describe the configuration. You use actions and commands to manipulate the data. Specifically, you open objects with the config command; you configure properties with the set command. Use the **delete** command to remove object configured settings from the configuration. The following table describes each of these elements.

Element	Description
Object	An *object* is a configuration container that contains properties of a specific configuration class. Objects are available at all configuration levels of the CLI; use the **config** command to open an object. Some examples of objects are: • vsp • dial-plan • enterprise
Property	A *property* is a value for a characteristic of an object. Properties are available at all configuration levels of the CLI; use the **set** command within an object to change properties. Some examples of properties are: • admin • apply-to-methods • domain-name

Element	Description
Command	A *command* is a tool used to change the configuration file. The changes do not affect AA-SBC until you save or update the configuration. Commands are available throughout all levels of the CLI. Some examples of commands are: • config • set • reset • move
Action	An *action* immediately acts on AA-SBC and effects one of the components. Actions are only available at the top-level prompt of the CLI. Some examples of actions are: • config save • dns-lookup

Editing objects and properties

The following table describes the commands used to edit the configuration. To enter configuration mode, enter **config** at the top-level command prompt Once in configuration mode, there are some commands and (and all actions) that are no longer available to you.

.

Command	Function

`config>` **`config`** *`object`*	Opens the specified object.
	Example: `config>` **`config box`**
	`config box>` **`config cli`**
	`config cli>`

`config` *`object>`* **`set`** *`property`*	Sets the specified property, either setting the value, overwriting a previous or default value, or adding an additional value. **Example:** config cli> **set display paged 24**

Command	Function
`config` *`object`*`>` **`delete`** *`object`*	Deletes your settings for the specified object from the current configuration as well as all objects (and their properties) contained within the deleted object. Anything that you can **config**, you can delete. For some services (e.g., NTP, SSH, web, etc.), **delete** kills the service. For some, **delete** returns the object to its default settings. The following example not only deletes the setting of the default-server (a property of the servers object), but also deletes all configured servers. **Example:** config servers> **delete sip-gateway NNOS-E-1**
`config` *`object`*`>` **`remove`** *`property`*	Removes the specified property from the configuration. You can remove properties that are references to other properties (see Referencing previously configured objects) and properties in a vector. (For properties that fit neither of these descriptions, you simply reset the value.) **Example:** config lcs test1> **remove domain-alias[2]**
`config` *`object`*`>` **`reset`** *`object`*	From the specified object, resets all of the properties to their default values. Note that an objects properties also include the properties of all subobjects. For example, resetting the VSP object resets the defaults of the enterprise, servers, and directories properties, as well as the accounting and location service properties and many more. **Example:** config vsp> **reset**

Saving changes to the configuration file

The method you use to save a configuration file effects which configuration file is modified. Saving also may move your position in the hierarchy. The following table describes each method of saving.

Command	Description
`config save`	Executed at the top-level prompt (**NNOS-E>**), saves the running config to either the saved config, or if a pathname is supplied, to that file name. You can choose standard, verbose, or XML formats. Standard format only outputs properties with a value different from the default; verbose outputs every property. The files are functionally equivalent.
`save`	Executed at the config-level prompt (**config>**), works the same as config save, above.
`commit (top)`	Executed from within an object (e.g., from **config vsp>** but not **config>**), saves changes from the working config to the running config and moves you to the top-level config prompt. You must still save changes to the saved config for them to be available at the next boot.
`return`	Executed from within an object, saves changes made in the current object to the working config and moves you up one level in the hierarchy. Changes do not get committed to the running config until you move to the top of the hierarchy.
`update`	Not directly accessible, writes changes from the running config to the saved config. The command appears when you try to exit config mode, and is executed by when you answer yes to the question: `Do you want to update the startup configuration (y or n)?` This prompt only appears if you have changed the running config.

Importing and exporting files in XML

To save the configuration to XML, select the XML format option when using the **save** or **config save** commands. You can then import this XML file to other devices to create a saved configuration. This will save you time if you have identical configuration settings across systems in the cluster. With XML, you can also work on the configuration file offline. In the CLI, XML and “standard CLI” config files are interchangeable, and the default save location, cxc.cfg (i.e. the startup config), is the same.

The following example saves config file cxc.cfg as XML:

```
config> save xml
```

Optionally, you can supply a file name after the format to save the configuration to a named file elsewhere.

Location in the CLI hierarchy

The CLI prompt always indicates where you are located in the CLI hierarchy. It does not show a complete object path hierarchy; instead it shows the object (and instance, if applicable) in which you are located. The following table describes the prompts that you can see.

Description	Prompt	Examples
Top-level prompt (default)	**NNOS-E>**	**Example:** NNOS-E> **show sessions**
Config-level prompt—object	**config>**	**Example:** `config>` **`config vsp`** `config enterprise>` **`config servers`**
Config-level prompt—object and instance	**config>**	**Example:** `config servers>` **`config lcs 1`** `config lcs 1>`

To see a complete object path hierarchy (with property settings) from your current location, use the **show** command:

```
config enterprise> show
vsp
 enterprise
  directories
  servers
  federations
  user-group-policy[1] grpEast "vsp\policies\session-policies\policy
   default"
  3pcc-servers

config directories> show
vsp
 enterprise
  directories
   admin enabled
   notes-directory abc
   phantom abc1
   on-failure ignore
   resolve-on-update false

config ldap XYZinc> show
vsp
 enterprise
  directories
   ldap XYZinc
    admin enabled
    tag test
    group East
    domain xyz.com
    host 0.0.0.0
    port 389
    transport TCP
    timeout 15000 ms
    username
    password-tag
    user-settings
    group-settings
    ignore-unresolved true
    ignore-domain true
```

Navigating the CLI

You can move through the object path hierarchy in a variety of ways. In addition, the CLI returns error messages to indicate the type of "transgression" it encountered at the command line.

Moving down through the hierarchy

You can move down through the CLI in two ways:

- by entering **config** commands individually, each on a new command line
- by entering the object path hierarchy on a single command line.

For example:

```
NNOS-E> config
config> vsp
config vsp> config enterprise
config enterprise> config servers
config servers> config sametime company1
config sametime company1>
```

Results in the same position as:

```
NNOS-E> config vsp enterprise servers sametime company1
config sametime company1>
```

Note that if you make a mistake in your object path entry, the system moves you to the last correctly completed object. For example:

```
config> config vsp enterprise goof
Invalid class
config enterprise>
```

Moving up the hierarchy

There are several commands that move closer to the top-level prompt. The following table describes the commands that allow you to navigate up though the CLI.

Command	Function
return	Moves to the previous level in the object hierarchy. If you are at the config> prompt, **return** is not available. You must use **exit** to back out to the NNOS-E> prompt.
commit or **top**	Moves to the top level configuration prompt (config>).
exit	Leaves configuration mode and returns to the NNOS-E> prompt. If you have made any changes to the configuration, you are prompted to commit changes before you exit.

Understanding CLI error messages

The following table explains the some of the more common messages that the CLI returns in response to navigation problems:

Message	Meaning
Invalid class	The object name you supplied either does not exist or is not available at this place in the CLI hierarchy.
Invalid command	You have entered a command name that does not exist at that point in the hierarchy. This could mean that you have attempted to issue: • a **set** when there are no properties to set • a **config** when there are no objects beneath the current container • a command that does not exist.
Invalid object	The instance of the object you tried to delete or display does not exist.
Invalid property	The property name you supplied either does not exist or is not available at this place in the CLI hierarchy.
Illegal value	The value you supplied for a property is not correct. This could occur if you entered the wrong type of value (e.g., entered “15” instead of “enabled”).
Required Property is Missing	You did not supply any value for an object or property that required one.
Too many arguments	You have tried to enter more values than the current property accepts.
Value is out of range	The value you supplied for a property is not within the allowable range.

Using the CLI

The following sections describe usage techniques for working with configurations at the command line.

Entering properties

The CLI properties, both required and optional, can be either a variable or one of multiple predefined values. The following example takes a *variable*, a value that you supply, such as 192.168.100.10:

```
set ip-address ipAddress
```

The following example takes a predefined value. Enter one:

```
set admin {enabled | disabled}
```

The following example takes both. If you select TLS for your transport protocol, you must enter the path to a certificate on the system:

```
set transport {TCP | TLS certificateReference}
```

Some objects have multiple, or compound, properties. For these, you can set more than one property for the object from the same command line configuration. In the following example, you must supply an IP address for the server, but you need not supply a transport protocol or a port, as these have default values:

```
set server ipAddress [UDP | TCP | TLS] [port]
```

It is important to note that properties are positional. If you want to set or change a property, you must supply any previous properties, even if they have default values. In the example above, if you wanted to change the port, you must first enter an IP address and transport protocol, even if you are not changing those values.

Displaying help text

There are several mechanisms for displaying help in the CLI. You can display a brief summary of the object or property you are setting. You can also display the list of available options from your current position.

Using the ? character with the config command

Use the question mark character (?) to display a brief summary of the objects or properties available to you. For example, to determine the type of servers you can configure, with a brief description of each, enter the following:

```
config servers> config ?
configure an object

 sametime       IBM Lotus Sametime Server
 lcs            Microsoft Live Communications Server 2005
 mcs            Nortel MCS
 avaya          Avaya PBX configuration
 sip-gateway    SIP Application Server or PSTN Gateway
 h323-gateway   SIP Application Server or PSTN Gateway
 sip-host       generic SIP source/destination
 dns-group      DNS resolved server group
 sip-connection SIP connection
```

Note that you receive the same result by simply typing **config** [ENTER] at the prompt.

Using the ? character with the set command

When you use the question mark with the set command alone, you display abbreviated help text associated with each property within the current configuration object:

```
config transport-policy 1> set ?

 Configures a transport layer DOS policy

 description
 admin               Sets whether resource is enabled or disabled
 select              Sets the properties in addition to remoteIP to
                         observe
 condition-list      Specifies conditional criteria on the database
                         search
 threshold           Sets the number of unique instances to be
                         considered a DOS attack
 period             Specifies how many seconds between database scans,
                         and how many seconds of data to analyze
```

If there are no properties to set, you receive an error:

```
config tls> set ?
Invalid command
```

Note that you receive the same result by simply typing **set** [ENTER] at the prompt.

When you use the question mark with a specific property name, and there are predefined values for the property (an enumeration), you display the values allowed for that property:

```
config transport-policy 1> set admin ?

 Sets whether resource is enabled or disabled

 enabled  Resource is active
 disabled Resource is inactive
```

If it is not an enumeration, you receive the simple help summary:

```
config transport-policy 1> set remote-ip-netmask ?

 Sets the mask of the remote-ip
config transport-policy 1>
```

Displaying available commands and properties

At any point in the CLI you can enter the question mark character to display available commands or options. Note that you can also type just the command and [RETURN] in some cases (as noted) to achieve the same result. When you use the verbose option (**-v**), the system displays the properties related to the objects being shown.

Note that when you display a list of actions, and in some cases show commands, the output is limited to the services registered with . Actions and status providers are only available to you if the service is registered (running). For example, if you do not have the authentication master service enabled, the RADIUS actions do not display. Therefore, an action that would have no effect does not appear as available.

The following table lists the commands available from each

.

Command	Function
`NNOS-E> ?`	Displays the list of actions and global commands available.

`NNOS-E> show`
`NNOS-E> show ?`
	Displays the set of valid show commands.

`config> ?`	Displays the list of CLI commands available at the top level of configuration mode.

`config> config`
`config> config ?`
	Displays the objects available for configuration from the **config>** prompt.

config *object*> ?	Displays the list of CLI commands available for the specific configuration object.
config *object*> **config** config *object*> **config** ?	Displays the objects available for configuration from the specific configuration object.
config *object*> set config *object*> set	Displays the list of properties available for configuration within the specific configuration object.

Displaying secondary commands

The CLI uses the concept of secondary objects and properties to filter out those items that are rarely used. These would be properties for fine-tuning a configuration, and would never be necessary for normal operations.

You cannot view secondary properties through the channels described in Displaying available commands and properties. When you list the available objects or properties, those that are secondary do not display, nor do their settings display in the standard help output. Instead, you must use the **help** or verbose **show** command to see the availability. You enter these properties as you would any other. However, command completion is not implemented for them.

The following example displays the standard properties of the DOS transport policy object (help descriptions removed for clarity):

```
config transport-policy test> set ?

 Configures a transport layer DOS policy

 description
 admin
 select
 condition-list
 threshold
 period
```

Compare the standard list to list available with the **help** command:

```
config transport-policy test> help
transport-policy
 description
 admin
 select
 remote-ip-netmask
 condition-list
 threshold
 period
 inactivity-timeout
```

The additional properties of **remote-ip-netmask** and **inactivity-timeout** are now viewable. To set these, use the standard procedure:

```
config transport-policy test> set inactivity-timeout 600
```

The secondary property that was manually set now appears in the regular **show** output:

```
config transport-policy test> show
vsp
 policies
  dos-policies
   transport-policy test
    description
    admin enabled
    select
    threshold 1000 instances
    period 30 seconds
    inactivity-timeout 0 days 00:10:00
```

Using the show command

When you use the **show** command from the config prompt, the system displays a list of configured objects in the running configuration If you specify the verbose option (-v), the system displays the properties related to the objects being shown. Note that this does not apply to show commands available from the top-level prompt.

At the top-level prompt or the config-level prompt, the show output includes all configured objects. The following example displays all configured objects in the running configuration:

```
NNOS-E> config show
cluster
box
services
master-services
vsp
external-services
preferences
access
features
NNOS-E>
```

The following example displays all configured objects and their associated properties:

```
NNOS-E> config show -v

cluster
 name NNOS-E-1
 box 1
  admin enabled
  hostname master
  timezone eastern
  name
  description Acme Packet Net-Net OS-E
  contact Jane Doe
  location Boston, MA
  identifier 00:55:66:00:11:22
  interface eth0
   admin enabled
   mtu 1500
   arp enabled
   speed 1Gb
   duplex half
   autoneg enabled
   ip a
    admin enabled
    ip-address static 192.168.100.100/24
---More---
```

You can also display just a portion of the running config, relevant to the object in which you are currently located. (This also shows you the path to your location.) For example:

```
config active-directory company1> show
vsp
 enterprise
  directories
   active-directory company1
    admin enabled
    tag east
    domain abcCo.com
    .
    .
    .
```

The following table summarizes the configuration display commands.

Command	Function
NNOS-E> **show** NNOS-E> **show ?**	Displays the set of valid status (show) commands.
NNOS-E> **config show** config> **show**	Displays a list of all configured objects in the running configuration.
NNOS-E> **config show -v** config> **show -v**	Displays, from the running configuration, a list of all configured objects as well as their associated properties.
config> **show *object*** config *object*> **show**	Displays, from the running configuration, the settings of the specified object, which includes immediate subobjects, and parentage, if applicable.
config> **show *object* -v** config *object*> **show -v**	Displays, from the running configuration, the contents of the specified object, all its properties (including the properties of subobjects), and parentage, if applicable.

Using command auto-completion

The CLI uses a command completion feature that automatically finishes typing an object or property name for you. Pressing the keyboard [TAB] or [SPACEBAR] executes the completion.

Note the following requirements for using command completion:

- You must type an entry until it is minimally unique on the command line before pressing [Tab] or [SPACEBAR]. If there are two commands that begin with the same spelling, the CLI cannot differentiate between the two until you type enough letters to distinguish one from the other.

Note: You must press [TAB] or [SPACEBAR] to complete the object or property name. It is not sufficient for the name to be minimally unique for execution.

- The entry must be a valid object or property in the hierarchy.
- The CLI does not auto-complete on user-configured instances or values.

The following example shows use of the auto-completion feature:

```
config box> con[SPACEBAR]nfig c[SPACEBAR]

possible completions:
 cli               CLI settings
 console           Console settings

config box> config co[SPACEBAR]nsole [ENTER]
config console>
```

In this example, pressing the [SPACEBAR] after typing con completes the config command line. However, pressing [SPACEBAR] after typing just **c**, which was not minimally unique, resulted in prompting for further characters. Entering the minimum unique characters, and then pressing [SPACEBAR] to complete the string, allows you to press [ENTER] to move to into console configuration mode.

If you are entering an object path, you can use the auto-complete feature for each component:

```
config> con[SPACEBAR]nfig v[SPACEBAR]sp en[SPACEBAR]

possible completions:
 enterprise Enterprise services
 enum       ENUM settings for phone number to URL conversion

config> config vsp en_
config> config vsp ent[SPACEBAR]erprise se[SPACEBAR]rvers
config servers>
```

Referencing previously configured objects

References allow you to re-use objects in the system. Therefore, you can define an object once, and then reference it later for other uses. For example, when configuring a DOS transport policy object, you need to include a reference to a condition list. Assuming you had configured a list named "remoteIP," you'd include it as follows:

```
config transport-policy> set condition-list vsp policies dos-policies
   transport-condition-list remoteIP
```

Entering references in the CLI

When you reference an object, you must use the full path name to the object. You can separate objects with backslashes or spaces. For example, either of the following is acceptable:

```
config transport-policy test> set condition-list
   vsp\policies\dos-policies\transport-condition-list 2

config transport-policy test> set condition-list vsp policies
   dos-policies transport-condition-list 1
```

However, in some cases, quotation marks are required if you are using spaces. This would be true when the configuration is a compound—it includes a reference and other values in a single property. In the example below, you must first specify the peer type (server) and then a reference to that server:

```
config source-route 192.168.100.100> set peer server “vsp enterprise
   servers sip-gateway pstn”
```

Note the following:

- when you are entering a reference, tab completion is available (unless the reference is within quotation marks).
- if you enter a path name to a reference that does not exist, the system creates an object of that name and supplies it with default values.

String requirements for the CLI

When a property or object requires a string (e.g., user name, directory service instance, etc.) the following rules apply:

- Any printable character is acceptable.
- If the string contains delimiters (white space or \ character), it must be enclosed in double quotes ““).
- If the string contains backslash character (\ ) and is therefore in quotes, you must use double backslash (\\ ) to get a single backslash in the result.
- Strings are case sensitive (i.e., admin is not the same as Admin).

The following string length limits are advised:

- object names up to 16 characters
- descriptions, usually in quotes, up to 32 characters
- regular expressions up to 128 characters.

Using regular expressions

A regular expression is a formula for matching strings that follow some pattern. Many of the conditions and predicates require a regular expression entry. uses PERL-compliant regular expressions.

You can configure replacement strings in several places throughout the configuration. The replacement string can include references to substrings from the source string. A substring in the source string is specified by enclosing it in parenthesis ( ). It is referenced in the destination string via \1 for the first substring, \2 for the second substring, and so on.

For example, if the source string is “The Quick Brown Fox and your expression is (.*)Quick(.*), and your replacement string is \Fuzzy\2, the result is “The Fuzzy Brown Fox”.

The following is a list of replacement string tokens along with examples. For each example, assume the input string is “The Quick Brown Fox”.

Replacement string token	Expression	Replacement	Output
\n The nth substring in the match.	“(.*)QUICK(.*)”	“\1Fuzzy\2”	“The Fuzzy Brown Fox”
\n\d The nth substring in the match followed by the digit d.	“(.*)QUICK(.*)”	“\Fuzzy\2\\7”	“The Fuzzy Brown Fox7”
\n\\d The nth substring in the match followed by the dth substring in the match.	“(.*)QUICK(.*)”	“\Fuzzy\2\\1”	“The Fuzzy Brown FoxThe”
\n\\\ The nth substring in the match followed by a single backslash character.	“(.*)QUICK(.*)”	“\1Fuzzy\2\\\”	“The Fuzzy Brown Fox\”

Replacement string token	Expression	Replacement	Output
\c An incrementing counter.	“(.*)QUICK(.*)”	`“\c\1\Fuzzy\2\c”`	`“1The 2Fuzzy Brown Fox3”`
\\ A single backslash character.	“(.*)QUICK(.*)”	“\\”	`“\”`

Use the **expression** action to develop and test regular expression match and replacement strings. For more information on this action see the expression description in Chapter 4.

Refer to one of the following sites for more complete instructions on writing regular expressions:

- http://www.perl.com/doc/manual/html/pod/perlre.html
- http://www.oreilly.com/catalog/regex/
- http://www.oreilly.com/catalog/regexppr/

Using relational operators

In policy building, uses some predefined relational operators for building conditions lists and predicate statements with elements of the same type. For example, use these operators to define ranges or compare values for equality or inequality. With them, your statements form logical expressions to determine choice, such as inclusion or exclusion, and sometimes action. (For enumerated lists, IP addresses, ports, and regular expressions, you use match and exclude statements.) The operators are as follows:

- eq=equal to
- ne=not equal to
- gt=greater than
- lt=less than
- ge=greater than or equal to
- le=less than or equal to

In addition, you can use match and exclude statements to define the use of the string. A match statement includes values that match the specified string; an exclude statement ignores them.

Setting time and time intervals

Several configuration objects and actions require that you set a time or time interval. The time specifies a date and time, for example, a start date. The interval reflects a number of days, hours, minutes, and seconds, for example, a refresh timer. The CLI accepts multiple entry formats for setting these intervals and displays them in the following formats:

```
master-services
 file-mirror
  external-backup
   admin enabled
   url
   refresh 0 days 00:30:00
```

You can enter a number of seconds; anything greater than 60 will be converted to *hh:mm:ss*. For example:

```
config external-backup> set refresh 120
config external-backup> show
master-services
 file-mirror
  external-backup
   admin enabled
   url
   refresh 0 days 00:02:00
```

You can enter minutes or hours explicitly. For example:

```
config external-backup> set refresh 10:30:00
config external-backup> show
master-services
 file-mirror
  external-backup
   admin enabled
   url
   refresh 0 days 10:30:00
```

To enter a number of days, enclose the string in quotation marks. You must enter the complete string for valid entry. For example:

```
config external-backup> set refresh “2 days”
Illegal value
config external-backup> set refresh “2 days 00:00:00”
config external-backup> show
master-services
 file-mirror
  external-backup
   admin enabled
   url
   refresh 2 days 00:00:00
```

In addition, the CLI accepts lexical representation for duration from the ISO 8601 extended format.

Using automatic values

Several properties within this object can be configured to allow to determine the appropriate value (a setting of **automatic**). The default value for these properties is automatically determined by based on the system hardware (processor, platform, memory, etc.). Although you can do so manually, do not change the value of these properties unless instructed to do so by Technical Support. Use the **show automatic-values** command to see the actual setting on your system.

Understanding passwords and tags

For increased security, uses a two-part password mechanism for passwords shared with other devices (also known as shared secrets). You must configure both a password and a tag. An enterprise or RADIUS server, for example, probably has a configured password that must use to access the server. This shared secret is the password. The tag is not the password itself, but rather a user-configurable name used to access the real password. By managing shared secrets, you can maintain the secrecy of the other passwords on other devices. An administrator can set up the tags and passwords; end users can work with the configuration files and use the password tag, without having access to the password itself.

For example, if the secret for your RADIUS server is **RadPswd**, you can create a secret-tag of **myTag**. When administrators configure to communicate with the RADIUS server, they supply the tag, **myTag**. The real password for server authentication, **RadPswd,** remains hidden to the user. The tag can be reused when creating other configurations that use the same real password. Or, if the password is compromised, it can be changed without changing the configuration on .

uses a password store to maintain the actual password known to the other device. Using a password store allows the shared passwords to be stored outside of, and not displayed in, the configuration file. Password tags are stored in the configuration.

Note: You can create a blank password by creating a tag without a corresponding password. This may cause problems when the external system, however, when it tries to authenticate .

This password mechanism applies only to cases of using a shared secret. It does not apply to passwords created for users under the **access** object. (These are stored as hashed data, never as plaintext.)

Using passwords and tags

There are several tag properties throughout the configuration. These include the various external databases, enterprise servers and directories, and phone configurations, among others. The minimum password length for users is set within the password-policy object. When setting the Linux root password, with the secret action, the default minimum-length of four characters is applied. No password length minimum is enforced for other secrets that live on other machines (RADIUS servers, etc.).

There are two ways to set up a password and tag correspondence—from within the object configuration and by executing an action.

In the example below, creates a password tag, **blue**, for an LCS server:

```
NNOS-E> config vsp accounting database group Boston server 1
config server 1> set password-tag blue
password: ***********
 confirm: ***********
config server 1>
```

Because the tag did not already exist, the system prompted for the real password. If the tag had previously been created, the system would have simply accepted the password tag as part of the configuration.

```
config server 1> set password-tag blue
config server 1>
```

You can also create a password and tag correspondence outside of the object configuration, using the **secret** action.

```
NNOS-E> secret set red
password: *****
 confirm: *****
Success!
```

If you re-execute the **secret set** action, and supply a different password, overwrites the password that was associated with the tag with the new password.

Use the **show secrets** command to display configured password tags:

```
NNOS-E> show secrets

tag
---
blue
red
```

Note: Passwords are maintained in a separate store; simply copying the configuration file between devices does not copy the password store. You can manually enter your passwords on each device. Or, can you use the secret **synchronize** action on the master device to copy your passwords on to other devices in the cluster.

Avoiding configuration conflicts with other users

To support two or more users editing the same copy of the configuration, implements a configuration conflict feature. This applies to all changes to the configuration, regardless of the tool used to make the changes (Management System, CLI, etc.). warns any user who attempts to update and save a configuration if the configuration has been saved elsewhere since it was loaded or last saved (indicating that the user does not have the most current version).

Note: When using web services to update the configuration, does not check for revision numbers, and therefore does not implement configuration conflict detection. When receives a setConfig message, it overwrites the running configuration regardless of whether unsaved changes have been made by other users or tools.

Revision management is the mechanism uses for avoiding these conflicts. Each top-level configuration object has an associated revision number. The top-level objects are: access, box, cluster, external-services, features, master-services, preferences, services, and vsp. increments the object revision number each time there is a saved change to that object (including one of its “children”). You can use the **show config-details** command to display the current revision number for each top-level object. Each time the box is restarted, the revision numbers revert to 1.

The example below shows sample output for a box that was newly updated, with changes made to the access and source-route objects. Because the vsp object is the top-level parent of the source-route object, increments the vsp count. A change that increments the revision can be an addition, modification, or deletion—anything that is then saved.

```
NNOS-E> config access
config access> delete permissions allowAll
config access> return
config> config vsp dial-plan
config dial-plan> config source-route src1
config source-route src1> set priority 500
config source-route src1> exit
Do you want to commit your changes before you exit (y or n)? y
Do you want to update the startup configuration (y or n)? y
NNOS-E> show config-details

object                 revision   size       changed
------                 --------   ----       -------
access                 2          572        08:31:08 Mon 2008-08-28
box                    1          6376       08:11:36 Mon 2008-08-28
cluster                1          9572       08:11:36 Mon 2008-08-28
external-services      1          151        08:11:36 Mon 2008-08-28
features               1          208        08:11:36 Mon 2008-08-28
master-services        1          558        08:11:36 Mon 2008-08-28
preferences            1          120        08:11:36 Mon 2008-08-28
services               1          925        08:11:36 Mon 2008-08-28
vsp                    2          40906      08:17:49 Mon 2008-08-28
```

If another user had made changes to any part of the configuration before the sample user had saved changes, the CLI presents the following message:

```
Do you want to commit your changes before you exit (y or n)? y
Your box changes will overwrite changes made by somebody else.
Are you sure that you want to commit your changes (y or n)? n
```

All configuration changes are recorded in the event log. Use the **show event-log** command to display the contents:

```
08:16:16 Mon 2008-08-28[notice] 1:manager[system] 'vsp' configuration
   changed by userA via console
```

Customizing the CLI display

You can use CLI commands to control the display of output to your screen and set your prompt.

Customizing the output display

You can configure the CLI to either output text a page at a time or scroll text continuously. You do this from CLI config mode (see the cli command description for more information). To scroll text:

```
NNOS-E> config
config> config box
config> config cli
config cli> set display scrolled
```

To pause the display with the `--More--` prompt, enter the following command, specifying the number of lines in your display:

```
config cli> set display paged 24
```

To temporarily change the display output without changing your configuration, you can execute the following command from the top level, using the **display** action:

```
NNOS-E> display paged 24
```

When you specify paged output, the `--More--` prompt accepts the following keystrokes:

Keystroke	Result
ENTER	Outputs the next line of text.
TAB	Outputs the remainder of the text.
ESC or Q or q	Cancels the display, outputs no more text, and returns to the prompt.
any other keystroke	Outputs the next page of text.

Resetting your prompt

By default, the system uses the prompt **NNOS-E>** as the top level prompt. If you'd like to change the prompt, use the following command:

```
NNOS-E> config box cli
config cli> set prompt "Think Big>"
config cli> exit
Do you want to commit your changes before you exit (y or n)? y
Do you want to update the startup configuration (y or n)? y
Think Big>
```

The prompt that you enter can be up to 64 alphanumeric characters. If you want the prompt to contain spaces, be certain to enclose it in quotation marks.

Exiting the CLI

The CLI supports several mechanisms for exiting hierarchy levels and the CLI itself. As you exit certain situations, you are prompted by the system as to whether you wish to commit changes. The following table describes the prompts and their implications:

Prompt...	Occurs when...	Responses result in...
Do you want to commit your changes before you exit (y or n)?	You have made changes to the working config and you are leaving config mode.	If you answer yes to the prompt, your changes are written to the running config (but not the saved config). They are used for your current session and until you next boot the system. If you answer no, your changes are discarded.
Do you want to update the startup configuration (y or n)?	You have committed changes from the working to running config, but have not yet saved them to the startup config for use when the system next boots.	If you answer yes, the changes are written to the startup config. If you enter no, the changes are not written, but are still in the running config. If you later answer yes to this question, without having rebooted the system, AA-SBC writes those changes (if they still exist in the running config) to the startup config.

The following table describes the commands used for exiting configuration mode levels and/or the CLI

.

Command	Function
`NNOS-E>` **`exit`**	From the top-level prompt, **exit** exits the CLI.
`NNOS-E>` **`quit`**	From the top-level prompt, **quit** exits the CLI.
`config>` **`exit`**	From the config prompt, **exit** exits configuration mode and returns to the top level CLI prompt. If you made changes to the configuration, the CLI issues a prompt asking you if you want to commit your changes.
`config` *`object`*`>` **`cancel`**	From within object configuration mode, **cancel** discards any changes to that object during the current running configuration session and moves you up one level in the configuration hierarchy.
`config` *`object`*`>` **`exit`**	From within object configuration mode, **exit** exits configuration mode and returns to the top level CLI prompt. If you made changes to the configuration, the CLI issues a prompt asking you if you want to commit your changes.

2. Global commands

Global command description

This chapter covers the global commands. A global command is a tool used to change the configuration file. The changes do not affect AA-SBC until you save or update the configuration. While global commands in general are available throughout all levels of the CLI, specific commands may only be available from certain prompts.

Note: Although it does not affect the configuration file, the **help** command is also described in this chapter.

Displaying global commands

At any level of the CLI, typing a question mark displays the options available to you from that point in the hierarchy. When at the top-level prompt, the global commands are mixed with the actions available. For example, in the list below, only the bolded **config** command is a global commands:

```
NNOS-E> ?
--More--
 clock                      set the system time
 cls                        clear terminal screen
 cluster                    cluster debug commands
 config                     configuration commands
 cpu-monitor                Monitor CPU usage; press Esc to cancel
 csta-moc-commands          Various commands related for MOC clients
 csta-uri-normalization     Perform CSTA URI normalization operations
--More--
```

Once you are in config mode, the actions are no longer available and AA-SBC only displays the relevant global commands:

```
config> ?

config        configure an object
delete        delete an object
dump dump     the configuration database
```

```
exit          exit configuration mode
help          display all configuration settings
save          save the running configuration
show          display configuration data
?             -v to show verbose help
```

Command summary

The following table lists and briefly describes the global commands and the level at which they are accessible.

Command	Description	Operates at this level:
`cancel`	Cancels changes to the current configuration object and restores the prior or default settings.	config *object*>
`commit`	Saves changes from the working config to the running config and moves you to the top-level config prompt.	config *object*>
`config`	Enters configuration mode or moves you to the next level down in the configuration hierarchy.	NNOS-E> config> config *object*>
`delete`	Deletes the specified configuration object from the running configuration.	config> config *object*>
`dump`	Displays a detailed summary of all the objects in the configuration database.	config>
`exit`	Leaves the current configuration mode and/or quits the CLI.	config> config *object*>
`help`	Lists all available commands and actions. Lists all possible objects for configuration from your current position in the configuration hierarchy.	NNOS-E> config> config *object*>
`move`	Moves items within a list (for use in objects where order effects processing sequence).	config *object*>
`quit`	Shuts down the CLI from the top level.	NNOS-E>
`remove`	Removes the specified property from the configuration.	config *object*>
`reset`	From the specified object, resets all properties to their default values.	config *object*>

Command	Description	Operates at this level:
return	Moves the configuration hierarchy up one level in the CLI.	config *object*>
save	Saves the current configuration file from the top level configuration mode.	config>
set	Applies a value to a specified property.	config *object*>
show	Displays the current CLI configuration or displays options at the current level of the hierarchy.	NNOS-E> config> config *object*>
top (config>	Saves changes to all open configuration objects and returns to the top of the configuration hierarchy.	config *object*>
top (NNOS-E>)	DIsplays by process by their CPU usage.	NNOS-E>

cancel

Purpose

Cancels all changes to the open configuration object, restores the prior or default settings, and exits that configuration object.

Prompt

```
config object>
```

Syntax

```
cancel
```

Example

The following CLI session configures Ethernet interface settings in the **box** configuration object; the **cancel** command ignores the new settings and reverts back to the prior or default settings.

```
config> config box
config box> config interface eth0
config interface eth0> set admin disabled
config interface eth0> set arp disabled
config interface eth0> cancel
config box>
```

commit

Purpose

Executed from within an object (e.g., from **config vsp**> but not **config**>), saves changes from the working config to the running config and moves you to the top-level config prompt. You must still save changes to the saved config for them to be available at the next boot.

This command is the same as the top (NNOS-E>) command.

Prompt

```
config object>
```

Syntax

```
commit
```

Example

The following example creates an IP interface, commits the change to the running config, and displays the new interface.

```
config> config box interface eth1 ip z
Creating 'ip z'
config ip z> commit
config> config box interface eth1
config interface eth1> show

box
 interface eth1
  admin enabled
  mtu 1500
  arp enabled
  speed 1Gb
  duplex full
  autoneg enabled
  ip d
  ip z
```

config

Purpose

Enters or moves deeper within AA-SBC configuration mode. The configuration mode provides access to all objects that configure and manage AA-SBC. You must have permissions for CLI access set to **normal** (standard CLI access) to use **config** and therefore change the configuration file. When you log into the AA-SBC Management System, the system automatically places you in the configuration mode.

Note that there are several configuration-related actions as well. See Chapter 3, “Actions” for more information.

Prompt

```
NNOS-E>
config>
config object>
```

Syntax

```
config
config objectName
```

Example

The following example illustrates entering configuration mode from the top-level prompt and then traversing the hierarchy two levels deeper.

```
NNOS-E> config
config> config vsp
config vsp> config enterprise
config enterprise> config servers
config servers>
```

delete

Purpose

Deletes your settings for the specified object from the running configuration as well as all objects (and their properties) contained within the deleted object. (Use the remove command to remove individual properties from an object in the configuration.) You delete an object from within the parent object. All setting associated with the deleted configuration object return to the system default settings. However, to restore the object and default settings to the configuration, you must enter configuration mode for that object.

Anything that you can **config**, you can delete. For some services (e.g., NTP, SSH, web, etc.), **delete** kills the service. For some, **delete** returns the object to its default settings.

Prompt

```
config>
config object>
```

Syntax

```
delete objectName
```

Example

The following example deletes all configured servers.

```
config enterprise> delete servers
```

The following CLI session displays the current Telnet settings followed by the **delete** command. The **config telnet** command restores the default Telnet settings to the running configuration.

```
config> config box interface eth0 ip x
config ip x> show
box
 interface eth0
  ip x
   admin enabled
   ip-address dhcp
   geolocation 0
   metric 1
   classification-tag
   security-domain trusted
   address-scope private
   filter-intf disabled
   telnet
config ip x> config telnet
config telnet> show
telnet
 admin disabled
 max-sessions 12
 idle-timeout 60 seconds
 port 22
config ip x> delete telnet
config ip x> show
box
 interface eth0
  ip x
   admin enabled
   ip-address dhcp
   geolocation 0
   security-domain trusted
   metric 1
   classification-tag
   address-scope private
   filter-intf disabled
config ip x> config telnet
config telnet> show
box
 interface eth0
  ip x
   telnet
    admin enabled
    max-sessions 8
    idle-timeout 600 seconds
    port 23
```

dump

Purpose

Displays a detailed summary of all the objects in the configuration database. This output is primarily for use by Technical Support personnel.

Prompt

```
config>
```

Syntax

```
dump
```

Example

The following is a segment of **dump** command output:

```
config> dump

object          class     id     obsrvrs    hldrs   weight
------          -----     --     -------    -----   ------
interface eth0    6       10        0         0     2601
interface eth1    6       33        0         0     1288
interface eth0    6       43        0         0      866
interface eth1    6       50        0         0      574
ip a              7       11        0         1     1149
ip b              7       27        0         1      708
ip c              7       30        0         1      628
ip d              7       34        0         1      750
ip a              7       44        0         1      798
ip b              7       51        0         1      506
ip a              7       55        0         0     1149
ip b              7       71        0         0      708
ip c              7       74        0         0      628
cluster          39       53        5         0     6214
messaging        42       52        0         0       58
messaging        42       70        0         0       58

--More--
```

exit

Purpose

Leaves the current configuration mode and/or quits the CLI.

The **exit** command does the following, depending on where you are in the CLI hierarchy. If entered:

- at levels within the configuration hierarchy (**config** *object*>), the **exit** command prompts you to commit changes to the current configuration. Type **n** (no) to discard all changes not previously saved and return to the top-level prompt. Type **y** (yes) to commit changes to the running configuration. You are then prompted to save these changes to the startup configuration. If you enter:
    - no, AA-SBC exits configuration mode and returns you to the top-level prompt. Changes are in the running config but not the startup config, and will be lost at the next system boot.
    - yes, AA-SBC saves changes to the startup config, exits configuration mode, and returns you to the top-level prompt.
- at the top level configuration mode (**config**>), returns you to the NNOS-E> prompt.
- at the NNOS-E> prompt, exits the CLI. (This command functions the same as quit).

Prompt

```
NNOS-E>
config>
config object>
```

Syntax

```
exit
```

Example

```
NNOS-E> config vsp
config vsp> set local-identity abcCo.com
config vsp> exit
Do you want to commit your changes before you exit (y or n)? y
Do you want to update the startup configuration (y or n)? y
NNOS-E>
```

```
NNOS-E> config vsp
config vsp> set local-identity abcCo.com
config vsp> exit
Do you want to commit your changes before you exit (y or n)? n
NNOS-E>

config> exit
NNOS-E>

NNOS-E> exit

\Connection to host lost.
c:\>
```

help

Purpose

Displays help that is dependent on your position in the hierarchy. From the top-level prompt (**NNOS-E>** by default):

- **help** provides a list of all available commands and actions, with brief text summaries.
- **help -v** (verbose help) provides the list of commands and actions with their possible settings (as well as text summaries).

From within configuration mode (either **config>** or **config** *object*>):

- **help** lists all possible objects for configuration, and their associated properties, from your current position in the hierarchy.
- **help -v** lists objects and properties from your current position in the hierarchy with brief text summaries.

Prompt

```
NNOS-E>
config>
config object>
```

Syntax

```
help [-v]
```

Example

The following example show a sample of each form of help.

```
NNOS-E> help
accounting         accounting commands
announce           Insert announcement on active call from a WAV file
archive            Run the archiving task for a given vsp
arena              Various arena debugging commands
arp                manage the ARP cache
assign-uri         Assign a sip URI to a user
--More--
NNOS-E> help -v

accounting                accounting commands
  copy     copy entries from one database to another
  database Examine an accounting database
   contact Contact a server
   create  Create the accounting table on a server
   count   Count the accounting records on a server
   query   Perform a query on a server
   clear   Clear a range of accounting records on a server
   reset   Reset the connections to a server
 announce                 Insert announcement on active call from a
   WAV file
 archive                  Run the archiving task for a given vsp
  specific Archive a specific set of sessions
   session Archive a single session
   between Archive a group of sessions between two times
--More--
```

move

Purpose

Re-orders an item to a different position in the hierarchy. For some properties, the order determines the sequence in which AA-SBC processes the properties.

For example, you may want to control the order in which AA-SBC checks user access. Initially, the order is determined by the order in which you configured the directories. Use the **show** command to verify the current order; the output displays an index inside a bracket. This is the number that you use in the move command.

```
config access> show

vsp
 access
  users[1]
  radius[2]
  enterprise[3]
```

The move command only appears when you have a list of items in which the order matters.

Prompt

```
config object>
```

Syntax

```
move item[originalPosition] destinationPosition
```

Example

The following example sets enterprise directory to be the first thing checked by AA-SBC:

```
config access> show

vsp
 access
  users[1]
  radius[2]
  enterprise[3]

config access> move enterprise[3] 1
config access> show
```

```
vsp
 access
  enterprise[1]
  users[2]
  radius[3]

config access>
```

quit

Purpose

Exits the CLI. The **quit** command only operates from the top-level of the CLI and has the same functionality as the exit command.

Prompt

```
NNOS-E>
```

Syntax

```
quit
```

Example

```
NNOS-E> config box
config box> set admin enabled
config box> top
config> exit
Do you want to update the startup configuration (y or n)? y
NNOS-E> quit

\Connection to host lost.
c:\>
```

remove

Purpose

Removes the specified property from the configuration. (Use the delete command to remove objects from the configuration.) You can remove properties that are references to other properties (see Referencing previously configured objects) and properties in a vector. (For properties that fit neither of these descriptions, you simply reset the value.)

For properties that accept multiple values, the system lists each configured value and assigns an index (inside a bracket) to that value. Supply the property name and the index value of the instance you want to remove.

Prompt

```
config object>
```

Syntax

```
remove propertyName[index]
```

Example

The following example shows an LCS server configuration with three domain aliases specified. (Multiple properties have been left out of the display for clarity). The **remove** command deletes one of the aliases and changes the index of the following alias.

```
config lcs test> show

vsp
 enterprise
  servers
   lcs test
    [PROPERTIES]
    domain-alias[1] abc.com
    domain-alias[2] lmn.com
    domain-alias[3] xyz.com
--More--
```

```
config lcs test> remove domain-alias[2]
config lcs test> show

vsp
 enterprise
  servers
   lcs test
    [PROPERTIES]
    domain-alias[1] abc.com
    domain-alias[2] xyz.com
--More--
```

reset

Purpose

Resets all object properties to the original default values. Note that properties also include the properties of all subobjects. For example, resetting the VSP object resets the defaults of the enterprise, servers, and directories properties, as well as the accounting and location service properties and many more. For those objects without default values (those that were created by their configuration), the **reset** command deletes the object and its subobjects.

Prompt

```
config object>
```

Syntax

```
reset
```

Example

The following example resets interface eth4 to its defaults:

```
config interface eth4> show
box
 interface eth4
  admin enabled
  mtu 1200
  arp enabled
  speed 1Gb
  duplex half
  autoneg enabled
config interface eth4> reset
config interface eth4> show
box
 interface eth4
  admin enabled
  mtu 1500
  arp enabled
  speed 1Gb
  duplex full
  autoneg enabled
```

return

Purpose

Moves you up one level in the configuration hierarchy and saves changes from that level to the running config.

Prompt

```
config object>
```

Syntax

```
return
```

Example

The following CLI session configures the administrative status of Ethernet interface eth0. The **return** command (executed twice) saves the change to the running config and moves back up the hierarchy to the top level configuration mode.

```
config> config box
config box> config interface eth0
config interface eth0> set admin enabled
config interface eth0> return
config box> return
config>
```

save

Purpose

Writes the running configuration to the default configuration file (/cxc/cxc.cfg), or if a path name is supplied, to that file name. You can choose standard, verbose, or XML formats. Standard format only outputs properties with a value different from the default; verbose outputs every property. By default, the configuration is saved in standard format.

You can also save your configuration file in XML format. You can then import this XML file to other AA-SBC devices to create a saved configuration. This will save you time if you have identical configuration settings across multiple devices in the cluster. With XML, you can also work on the configuration file offline. In the CLI, XML and “standard CLI” configuration files are interchangeable, and the default save location, cxc.cfg (i.e., the startup config), is the same.

AA-SBC creates a numbered backup (cxc.cfg.#) with each execution, creating up to four backups. Files are saved in the /cxc/backup directory.

This command works the same as the config **save** action.

Syntax

```
save [standard | verbose | xml} [fileName]
```

Example

The following example saves the running configuration to standard CLI script:

```
NNOS-E> config
config> config box cli
config cli> set display scrolled
config cli> top
config> save
```

set

Purpose

Configures properties of an object. It either sets the value, overwrites a previous or default value, or adds an additional value.

Prompt

```
config object>
```

Syntax

```
set propertyName
```

Example

The following example sets properties of a box:

```
NNOS-E> config box
config box> set admin enabled
config box> set timezone eastern
config box> set name NNOS-E-1
config box> set contact “Jack Spratt”
```

show

Purpose

Displays configuration entries and status reports for each system provider. To view a listing of the **show** commands that are available from a particular point in the command hierarchy, navigate to the command mode and enter:

show ?

To view the filter fields available for specifying an entry to display, enter a question mark after the **show** *argumentName* entry. For example:

show dial-plan ?

Using the show command also indicates your command path, because it displays the path from the top of the hierarchy to your position.

See Chapter 4, “Status provider show commands”, for a detailed description of use of the **show** command and a description of each status provider report. See Displaying help text for more details on using the **help** command.

Prompt

```
NNOS-E>
config>
config object>
```

Syntax

```
show [objectName]
```

Example

The following examples shows different points from which you can enter the **show** command and different types of output:

```
NNOS-E> show ?
show commands
accounting-database         request information for accounting database
   connections
accounting-process         General statistics from the accounting
   process
accounting-recent          calls recently accounted
accounting-server          request information for accounting servers
accounting-status            accounting activity information
actions                      action provider statistics

--More--

NNOS-E> show interfaces

interface     name     ip-address            op-state      type
---------     ----     ----------            ---------     -----
eth0          a        192.168.215.100/24     up           public
eth0:1        b        192.168.215.110/24     down         public
eth0:2        c        192.168.215.120/24     up           public
eth1          d        192.168.216.100/24     up           public

NNOS-E> config
config> config box
config box> config cli
config cli> show

box
 cli
  prompt NNOS-E>
  banner Shutdown at 12:00 midnight
  display paged 24
```

top (config>

Purpose

Saves changes from the working config to the running config and moves you to the top-level config (**config**>) prompt. You must still save changes to the saved config for them to be available at the next boot. This command is only available from within an object.

This command is the same as the commit command.

Prompt

```
config object>
```

Syntax

```
top
```

Example

The following example creates an IP interface, commits the change to the running config, and displays the new interface.

```
config> config box interface eth1 ip z
Creating 'ip z'
config ip z> commit
config> config box interface eth1
config interface eth1> show

box
 interface eth1
  admin enabled
  mtu 1500
  arp enabled
  speed 1Gb
  duplex full
  autoneg enabled
  ip d
  ip z
```

top (NNOS-E>)

Purpose

Displays the processes that are top users of a category of system resources. You can select the category to sort on by typing the first letter of the name (except for Process Name, type N). The system resorts the list of processes, in descending order and at two-second intervals, based on their CPU use. Press ESC to exit the display. This command is only available for users with debug permissions.

Prompt

```
NNOS-E>
```

Syntax

```
top
```

Example

```
NNOS-E> top
Procs:  75  Interval: 2  CPUs: 4  Mem:  4053M Swap:   0M  Kern: 12%   CPU:  0%
-------------------------------------------------------------------------------
  PID  Process Name      CPU %  Memory  Resident  Locked  Threads   Files
-------------------------------------------------------------------------------
 2976  managerl.elf       0.1%    180M       38M      0K       41      69
 4232  login              0.0%      1M      352K      0K        1       4
 4230  sshd               0.0%      4M        1M      0K        1       9
 4216  postmaster         0.0%     15M       11M      0K        1      24
 4215  postmaster         0.0%     15M       12M      0K        1       6
 3928  java               0.0%    581M      108M      0K       35      12
 3927  java               0.0%    456M      130M      0K       35      14
 3926  authl.elf          0.0%     42M       27M      0K       24      38
 3924  java               0.0%    450M      120M      0K       48      13
 3864  postmaster         0.0%     15M        4M      0K        1       5
 3858  postmaster         0.0%     15M        4M      0K        1       5
 3857  postmaster         0.0%     15M        4M      0K        1       5
-------------------------------------------------------------------------------
       Totals:            0.1%   3971M     1745M      0K      494     543
```

3. Actions

Actions description

This chapter covers AA-SBC actions. An action immediately acts on AA-SBC and effects one of the components (manipulates data), whereas objects and properties describe the configuration. Actions are only available at the top-level prompt of the CLI (or through the AA-SBC Management System **Actions** tab).

Most actions become available when AA-SBC starts, but some may only become available when the corresponding service or provider registers with AA-SBC. The registration is dependent on the configuration. For example, the **directory-reset** action cannot register if the directory service is not configured.

accounting

Purpose

Sets up and debugs a remote accounting database. These are the databases identified with the **accounting** database object.

- **contact**—contacts the specified server to verify connectivity.
- **create**—creates the accounting table on the specified server. You can select, also, whether to execute the command. If true, AA-SBC executes the command and creates the table. If false, AA-SBC returns the SQL statement to create the table, but it does not actually create it. Use this, for example, to create the table using another tool.
- **count**—returns a count of the number of accounting records on the specified server.
- **query**—queries the server using any SQL query you enter. Optionally, define the number of rows you'd like to query.
- **clear**—deletes a range of accounting records from the specified server. Define the range using the format *hh:mm:ss yyyy-mm-dd*. If you do not enter a range, AA-SBC deletes all records.
- **reset**—resets the connection to the specified server.
- **flush**—flushes an accounting target
- **purge**—forces an immediate run of the purge process and cleans up all CDRs on the file system that are eligible for deletion. See the **purge-criteria** property of the accounting object for information on eligibility.

Syntax

```
accounting database contact accountingServerReference
accounting database create accountingServerReference [true | false]
accounting database count accountingServerReference
accounting database query accountingServerReference query [rows]
accounting database clear accountingServerReference [from] [to]
accounting database purge
```

Example

In the following examples, the actions first check connectivity to the server and then try to create a table in the database. The error indicates that the table already exists. The final example illustrates a query.

```
NNOS-E> accounting-database contact vsp accounting database group
   Boston server Boston
Success!
NNOS-E> accounting database create “vsp accounting database group
   Boston server Boston”
ERROR: relation “acctcallstruct” already exists
Failed to create table on server
NNOS-E> accounting database query “vsp accounting database group
   Boston server Boston” “select count (*) from acctcallstruct”
<ResultSet><header>count</header><row><ResultSetRow><column>0</
   column></ResultSetRow></row></ResultSet>
```

accounting flush

Purpose

The accounting flush action allows you to override existing configuration settings to manually purge accounting files. The results of this operation are logged and you can view the logs to obtain more information on files that have been flushed. These are the targets identified with the accounting object that can be specified:

- file-system—A rollover is performed when the flush is executed. This means the current file is closed and a new file has automatically been created. In the case of a postgresql file, the trailer is written to the temp file and renamed in the proper format.
- external-file-system—A file whose send has failed previously retries the flush immediately when this action is executed before waiting for the retry interval to try again.

Syntax

```
accounting flush file-system url
accounting flush external-file-system path
```

Example

```
NNOS-E> accounting flush file-system “vsp\accounting\file-system\path
    a”
  Accounting flush has been initiated...check the event log for
    results.
NNOS-E>
```

accounting reapply

Purpose

Re-exports accounting entries from one target to another. Use this, for example, if the connection is broken or in any event that prevented accounting records from being written to the external target. When you execute this action, AA-SBC returns the qualifying records on the file system to an unprocessed state. As a result, the records are resubmitted to the selected accounting targets. This action is limited to data that falls within the time frame set with the **accounting** object **retention-period** property.

Enter the begin and end time of the period for which you want to copy records. Enter the times in the format *hh*:*mm*:*ss yyyy-mm-dd*. If you do not enter the full format, the system completes the entry with the current date. For example, if on April 16 of 2007 you enter simply “1:00,” the system uses 01:00:00 2007-04-16. Also enter a reference to one or more previously configured targets.

Syntax

```
accounting copy startTime endTime [databaseReference]
    [syslogReference] [radiusReference] [filesystemReference]
```

Example

```
NNOS-E> accounting reapply 1:00:00 2008-09-01 00:00:00 2009-09-02 “vsp
    accounting database group myDB”
Success!
NNOS-E>
```

add-device

Purpose

Mounts or remounts data drives on to the system. You may want to use this in cases when you do a system upgrade or if you are moving a hard drive from one AA-SBC device to another. When upgrading software, data drives will no longer be mounted after the upgrade. Run this action after the restart to restore the data drives to operate with the new software. Specify the drive to add and the file system in use by that drive.

Note that the format action automatically performs the equivalent of the add-device and mount actions. If you have data on a drive that you want to maintain, be sure to manually execute these two actions. Do not use format as it will remove all data from the drive.

Syntax

```
add-device {data-1 | data-2} {reiser-3 | xfs}
```

Example

```
NNOS-E> add-device data-2 xfs
Success!
```

announce

Purpose

Plays the specified .WAV file on a connected/anchored call. Use **show active-calls** (the **sessH** field) to retrieve the session handle. You can specify in which direction to insert file, either in the inbound direction (for the caller), outbound (for the callee), or both. Additionally, you can specify when the call should terminate. If termination is set to **true**, AA-SBC hangs up the call after playing it. By default, AA-SBC does not hang up the call (terminate set to **false**). (By contrast, the file-play action establishes a call and hangs up after the .WAV file is finished playing.)

Syntax

```
announce sessionHandlefile filename [in | out | both] [true | false]
```

Example

```
NNOS-E> show active-calls

Active Calls:
----------------------------------------------------------------------
SessionID: 04c298ee0bdbe1f6
From: “rick” <sip:3933@barry.acmepacket.com>;tag=102949606215911
To: <sip:1234@barry.acmepacket.com>
CallID: 34990D8B-7B2F-4E47-BFD3-BA1808A6835E@172.30.1.6
State: B2B_CONNECTED
sessH: E33E6C12
Connect:
Duration: 12 seconds
In Conn:
Out Conn:
----------------------------------------------------------------------
Total Active Calls: 1

NNOS-E> announce 0xE33E6C12 /cxc_common/media/greeting.wav both
Success!
```

archive

Purpose

Saves stored sessions for the specified VSP. The archiving action archives all data that has not been successfully archived previously. You can archive a specific session or sessions that occur between specified times using the archive specific action.

Use this action to initiate the backup immediately; use the task object to schedule automated backups. You must enable archiving with the archiving object for this action to succeed. Use the **show archive-result** command to view the outcome of archiving operations.

Note: If you have record-based archiving configured, manual archive operations will fail. You must set the **record-count** property of the archiving object to 0 for this archive action to work.

If do not enter a VSP name, AA-SBC archives the default VSP.

Syntax

```
archive [VSPname]
```

Example

```
NNOS-E> archive
The specified vsp is not configured for archiving
NNOS-E> config vsp accounting
config accounting> config archiving
config archiving> set admin enabled
config archiving> top
config> exit
Do you want to update the startup configuration (y or n)? y
NNOS-E> archive
Success!
NNOS-E>
```

archive specific

Purpose

Saves specific sessions for later retrieval. You can archive either a specific session, or a range sessions that occur between specified times. (Use the archive action to archive all sessions.) You must enable archiving with the archiving object for this action to succeed.

When you execute the action, AA-SBC creates temporary files based on the session ID, and writes them to either the file displayed in the response message (**session**) or the specified directory (**between**). Once the files are archived to AA-SBC, you can move the file(s) off the device (using TFTP, PSCP, etc.).

Select one of the following operations:

- **session**—saves a specific session, identified by its session ID number, to a system-assigned temporary file name. Or, you can specify a file name. Use the AA-SBC Management System **Call Logs** page to determine the session ID. In the CLI, there are various status providers that display the session ID as part of their output.
- **between**—saves all sessions in the specified time range to a file. Enter a time in the format *hh*:*mm*:*ss yyyy-mm-dd*. If you do not enter the full format, the system completes the entry with the current date. For example, if on April 16 of 2006 you enter simply “1:00,” the system uses 01:00:00 2006-04-16. In addition, you must enter a directory to which the files can be written.

Syntax

```
archive specific {session sessionID [fileName] | between startTime
   endTime directory}
```

Example

```
NNOS-E> archive specific between 8:00 22:00 /nightly_archives
Success!
NNOS-E> archive specific session 0x04C20B07E06BD88B
/tmp/arc41254.zip
NNOS-E>
```

arp

Purpose

Manages the ARP cache.

- **delete**—Removes either a specific IP address or, if no address is specified, all entries from the ARP cache. Use the **show arp** command to view the contents of the cache.
- **request**—generates an ARP request directed to the specified IP address. This option is analogous to the ping action; both verify connectivity to an address.
- **reply**—generates an ARP reply for an interface (also known as a gratuitous ARP). The **arp reply** action broadcasts the specified interface to all hosts in the network. Use this to force a broadcast notifying of a change to the interface.

Syntax

```
arp {delete [ipAddress] | request ipAddress [ethX] | reply ethX}
```

Example

```
NNOS-E> show arp
ip-address      type    flags               mac-address      interface
----------      ----    -----               -----------      ---------
172.26.0.1      ETHER    COM                  00:13:1a:73:d1:c2 eth0
172.26.0.189    ETHER    COM                  00:04:23:b2:fa:8a eth0
172.26.0.199    ETHER    COM                  00:04:23:b5:3e:60 eth0
172.26.0.252    ETHER    COM                  00:04:23:b2:f6:c6 eth0

NNOS-E> arp request 172.26.0.252
172.26.0.252 is 00:04:23:b2:f6:c6

NNOS-E> arp delete 172.26.0.252
Success!

NNOS-E> arp request 172.26.0.252
Did not receive an ARP response

NNOS-E> show interfaces
interface  name            ip-address         op-state         type
---------  ----            ----------         --------         ----
eth1       b               192.168.216.210/24 up               public
eth1.5     one             192.168.216.200/24 up               public

NNOS-E> arp reply eth1.5
Success!
```

auth request

Purpose

Tests validity of a variety of different authentication types. Note that this command tests a RADIUS group; to test credentials on an individual server, use the radius **test** action.

.

Note: This command is available for the CLI only.

Enter the following:

- **-t**—Specifies authentication type to test, which can be one of the following:

  Local—Perform local authentication

  RADIUS—Perform RADIUS authentication

  DIAMETER—Perform DIAMETER authentication

  Directory—Perform Directory authentication

  Accept—Accept all authentication attempts

  Reject—Reject all authentication attempts
- -g Configuration reference (“vsp\radius-group Boston”, etc. Overrides -t.

  For RADIUS and DIAMETER, must specify a group, not a server.
- -n—User name
- -p—Password
- -c—Request count. Default is 1 request.
- -r Rate, in requests/second. Default is no delay between requests.
- -u User name/password authentication. This is the default.
- -d Digest authentication. Default is user name/password authentication.

Actions

- -dr Digest realm. Default is 'testrealm'; implies '-d'.

-dm Digest method. Default is 'INVITE'; implies '-d'.

-du Digest URI. Default is 'sip:5555551212@example.com'; implies '-d'.

-q Quiet mode.

Syntax

```
auth request -t [type] -g [config reference] -n [name] -p [password] -c
   [count] -r [rate] -u -d -dr [realm] -dm [method] -du [URI] -q
```

Example

```
NNOS-E> auth request -g “vsp\radius-group East” -n user1 -p pswd1

Provider type:     RADIUS
Config reference:  vsp\radius-group East
User name:         user1
Password:          pswd1
Request type:      Password
Request count:     1
  Rate:            0/second
  Period:          0 seconds


Initiated     1 requests.
Received      1 successes in 0.002 seconds (500.0/second).
                Min 0.002, Average 0.002, Max 0.002 seconds
Received      0 failures.
```

authentication-cache-flush

Purpose

Removes all entries from the AA-SBC authentication cache, used for re-authenticating REGISTER requests.

Syntax

```
authentication-cache-flush
```

Example

```
NNOS-E> authentication-cache-flush
```

```
Success!
NNOS-E>
```

autonomous-ip

Purpose

Reports whether two endpoints are within the same autonomous-ip-group. Use this action to test your autonomous IP configuration and connectivity. Note that when supplying fields for this action, you must supply all values that are configured for the endpoint. Otherwise, evaluation results will be inaccurate. For example, if the source endpoint has an associated routing-tag, you must supply that value for the *srcTag* field. Enter the following values, as applicable:

- **address**(required)—specifies the public IP address for the endpoints. This is the address of the firewall (e.g., router) or, if directly connected, the endpoint itself.
- **public-ip**—specifies the private IP address for the endpoints. If sip **nat-translation** is enabled, this is the IP address of the phone in the private network. If **nat-translation** is disabled, this value should be 0.0.0.0 (the default).
- ***srcTag*** and ***destTag***—specifies the routing tag associated with the endpoint. This is a tag derived from either the ip **routing-tag** property for an interface or the routing-settings **ingress-** and/or **egress-classification-tag** in the session configuration.

The results of this action are either release or anchor. Release indicates that the endpoints are both part of a group and therefore need not be anchored. Anchor indicates that they are not part of a group so AA-SBC must anchor the call media.

Syntax

```
autonomous-ip-evaluate srcPublicIP destPublicIP [srcPrivateIP]
    [destPrivateIP] [srcTag] [destTag]
```

Example

```
NNOS-E> show autonomous-ip-group

group-name    gateway    connected selfConnected
----------    -------    --------- -------------
group-1                  true      true
group-2                  true      true
```

Actions

```
NNOS-E> show autonomous-ip-route

name           match             hits
----           -----             ----
group-1        10.10.10.0/24     4
group-2        192.168.1.0/24    4
NNOS-E> autonomous-ip-evaluate 10.10.10.5 192.168.1.5
Result is anchor

NNOS-E> autonomous-ip-evaluate 10.10.10.5 10.10.10.6
Result is release
```

bandwidth-calculate

Purpose

Calculates the amount of bandwidth required for a single RTP stream. You can change the overhead of the optional fields to more closely match your network scenario, and use the output for network planning. Use the s**how codec-info** status provider to determine the ptime and payload for the CODEC in use. Enter the following fields:

- ***packetInterval***—enter the ptime for the CODEC in use.
- ***rtpPayload***—enter the bytes of RTP payload per packet.
- ***rtpOverhead***—enter the bytes of RTP overhead per packet. Additional overhead might be the result of SRTP authentication or MKI.
- ***ipOverhead***—enter the bytes of IP overhead per packet. Additional overhead might be the result of running IPsec or other IP options.
- ***ethOverhead***—enter the bytes of IP overhead per packet. Additional overhead might be the result of using VLAN tags.

Syntax

```
bandwidth-calculate packetInterval rtpPayload [rtpOverhead]
   [ipOverhead] [ethOverhead]
```

Example

```
NNOS-E> bandwidth-calculate1
Location at public:192.168.10.1 and private:0.0.0.0 is associated with
   group test
NNOS-E> autonomous-ip-lookup uri sip:00004e20@acmepacket.com
URI sip:00004e20@acmepacket.com is associated with group test
```

base-64

Purpose

Encodes a data string into base-64 or decodes a data string from base-64. You can select to encode a hexadecimal or text string or decode a base-64 string into hex or text. In other words, the encode option indicates how to treat the input (as hex or a string of characters). The decode option specifies the output type.

Syntax

```
base-64 {encode-hex | encode-text | decode-hex | decode-text} data
```

Example

```
NNOS-E> base-64 encode-hex 0x1234567890
EjRWeJA=
NNOS-E> base-64 decode-hex EjRWeJA=
0x1234567890
```

call-control

Purpose

Sets up and manages calls that are originated via AA-SBC. To do this, you must first create the call using the **call-control call** action. That action results in AA-SBC creating a handle for the call. That handle is then used to identify the call in all further call control actions. Use the following actions to control a call:

- **call**—creates a call. Specify the destination (to) and originating (from) SIP or TEL URI. This action results in a call handle. Optionally you can enter any of the following fields:
  - optionally enter a request ID. This value is returned in any generated events for the call.
  - specify whether to ring originator first (**enabled**, the default). I **disabled**, AA-SBC rings the terminator first.
  - specify whether to place the call asynchronously. If **enabled**, AA-SBC returns an action response before the call is connected. If **disabled**, the default, the action does not return until the call connects.

Actions

  - specify a transport protocol or use the default of **any**.
  - reference a saved session configuration to apply to the call.
- **hold**—places the call on hold. Specify the AA-SBC-created handle.
- **retrieve**—reactivates a call that was placed on hold. Specify the AA-SBC-created handle.
- **transfer**—transfers a call to a new destination. Enter the handle to identify the call and a SIP or TEL URI to identify the destination.
- **disconnect**—ends an active call. Specify the AA-SBC-created handle of the call.
- **join**—performs an “announced transfer” of a call to another call. For example, if you have a phone that has placed a call to two different destinations, AA-SBC assigns each its own handle. You can then join the two calls, letting the original originating phone drop out.
- **loop**—places a loopback call, which loops back RTP media, allowing AA-SBC to gather call quality statistics.
- **annotate**—attaches an annotation (text description) to the call identified by the specified call handle. Use the **get-annotation** option to retrieve the text on AA-SBC. This data is also available to other systems.
- **get-annotation**—retrieves and displays an annotation (text description) attached to the call identified by the specified call handle. Use the **annotate** option to attach the text.
- **park**—places a call to the specified endpoint (a single-sided phone call). Using that call handle at a later time, you can use the **connect** option to connect it to another endpoint. Optionally, specify whether to perform this action asynchronously (see the **call** option for a description of asynchronous). You can also reference a saved session configuration to apply to the call.
- **connect**—Using the call handle of a previously parked call, connect that call to the specified endpoint. Optionally, specify whether to perform this action asynchronously (see the **call** option for a description of asynchronous). You can also attach a request ID to be included in related events.
- **terminate**—disconnects the specified call. The other end of the call is left on hold.

- **memo-begin**—begins recording a voice memo, which is saved as a WAV file to the specified file name. Use this with the **play** option. The system records anything spoken into the phone until the memo-end action occurs or the phone hangs up.
- **memo-end**—ends recording of a voice memo that was started with the **memo-begin** option. Until you end the memo, the WAV file created cannot be used.
- **play**—plays the indicated WAV file on the call specified by the call handle.
- **drop-file**—plays the indicated WAV file on the call specified by the call handle, and parks the originator.
- **notify**—sends the specified SIP notify message to the endpoint.
- **message**—connect to an endpoint, play a file, and terminate the call.

Syntax

```
call-control call to from [requestId] [enabled | disabled] [enabled |
   disabled] [any | UDP |TCP |TLS] [SessionConfigReference]
call-control hold handle
call-control retrieve handle
call-control transfer handleto
call-control disconnect handle
call-control join handle1 handle2
call-control loop handle
call-control annotate handletext
call-control get-annotation handle
call-control park endpoint [enabled | disabled]
   [SessionConfigReference]
call-control connect handle endpoint [enabled | disabled] [requestId]
call-control terminate handle
call-control memo-begin handle filename
call-control memo-end handle
call-control play handle filename
call-control drop-file handle filename
call-control notify handle event
call-control message filename endpoint [from] [requestID] [enabled |
   disabled] [sessionConfigReference]
```

Example

The example shows placing a call, and the resulting handle—3774756626. Next the call is placed on hold, retrieved, and then disconnected.

```
NNOS-E> call-control call sip:4135550002@rv.com sip:4135550001@rv.com
   any “vsp session-config-pool entry manage”
```

Actions

```
3774756626

NNOS-E> call-control hold 3774756626
3774756626

NNOS-E> call-control retrieve 3774756626
3774756626

NNOS-E> call-control disconnect 3774756626
3774756626

NNOS-E> call-control disconnect 3774756626
specified instance does not exist
```

call-failover

Purpose

Flushes the call-failover database of any signaling and media-session records used to maintain call state between redundant AA-SBC devices. Typically this action is used for troubleshooting purposes only, when advised to do so from customer support.

This action is only available in the CLI if you have enabled the call-failover master service. You can also schedule this action as part of routine maintenance using the **task** object.

Syntax

```
call-failover flush
```

Example

```
NNOS-E> call-failover flush
Success!
```

call-lookup

Purpose

Displays, for a specified Request URI, the dial-plan settings that AA-SBC assigned, the routing arbitration process, and the selected server.

This action simulates the call routing path, but does not actually trigger an outbound call. It exercises both dial plan and location database lookups. Its output indicates which dial plan entry (or location cache entry) the call would use and the next hop.

Enter a Request URI. The action returns results for any AOR that matches a configured dial plan, or you can find the URI of interest in the AOR field displayed with the **show location-cache** command.

Syntax

```
call-lookup requestUri [wildcard | named tag | anonymous] [soureceIP]
   [localPort] [fromURI] [toURI] [contactHeader]
```

Example

```
NNOS-E> call-lookup sip:2078548355@elmaple.com
Resulting priority 100 sequential hunting total 1 next 0
option 0: preference 10000 bandwidth 0 cap 0 rate 0 QOS null selected
   192.168.77.179
All matching routes:
route domain elmaple.com priority 100 best yes
This call will be forwarded to 192.168.77.179 transport UDP port 5060
```

call-lookup-detail

Purpose

Displays, for a specified Request URI, the content of the session configuration associated with the selected dial-plan. If the SIP URI matches a dial-plan (source-) route or arbiter, the action returns the session configuration for that entry (or the default session configuration if the entry does not have one assigned). If the dial-plan **route** has the **peer** set to server, and the server has a session configuration, this action displays the merged session configuration. (The server session configuration takes precedence over the route session configuration.)

Syntax

```
call-lookup-detail requestUri [wildcard | named tag | anonymous]
   [soureceIP] [localPort] [fromURI] [toURI] [contactHeader]
```

Example

```
NNOS-E> call-lookup-detail sip:2078548355@elmaple.com
Resulting session config merged from server/group egress:
```

```
sip-settings
mode auto-determine
transport any
port auto-determine
route-hdr none
route-hdr-use-fqdn enabled
route-hdr-uri-host
route-hdr-add-register-msg disabled
route-hdr-preprocess-strip disabled
lcs-compatibility disabled
in-server unknown
out-server unknown
utilize-contact enabled
add-contact-nat disabled
compress-signaling disabled
preserve-call-id disabled
preserve-cseq disabled
proxy-generate-100-trying
handle-3xx-locally enabled
handle-3xx-locally-lookup-original-invite disabled
session-timeout 300 seconds
session-duration-max 0 seconds
--more--
```

cert-gen

Purpose

Generates a 1024-bit key pair (public and private) using the RSA algorithm. In addition, the action generates an X.509 V3 self-signed certificate. You can subsequently use the keyfile you create to generate a Certificate Signing Request (CSR) to send to a CA. (Generate the request using the cert-request action and update the self-signed certificate with the validated certificate using the cert-update action.)

The following fields must or can be specified as part of the action to create a certificate's Distinguished Name (DN), which uniquely identifies the entity:

- ***keyfile***—the name of the file to which the key and certificate will be saved.
- ***password***—the password used to encrypt the private key.
- ***alias***—the name given to this entry within the keyfile.
- ***commonName***—the fully-qualified domain name for the site using the certificate. To include subdomain, use a wildcard (e.g., www*.companyABC.com).

- **[*daysValid*]**—the number of days that you are requesting from the CA that your certificate remains valid. The default is 365 days.
- **[*country*]**—the two-letter ISO country code for the country in which the business is registered. See the *International Organization for Standardization* for a listing of codes.
- **[*alternateName*]**—any other name you want associated with the distinguished name (e.g., an email address).
- **[*organization*]**—the legally registered name of the business or holder of the domain name.
- **[*organizationalUnit*]**—a division within the organization (e.g., marketing, engineering).
- **[*state*]**—the name of the state, province, region, or territory in which the business is registered. Do not abbreviate this field.
- **[*locality*]**—the name of the city or locality in which the business is registered. Don not abbreviate this field.

When you invoke this action, you are required to enter and confirm your password. Enter the password that you used, in this action, to create the key pair.

See *RFC 1779, A String Representation of Distinguished Names*, for more information.

Syntax

```
cert-gen keyFile password alias commonName [daysValid] [country]
   [alternateName] [organization] [organizationUnit] [state]
   [locality]
```

Example

```
NNOS-E> cert-gen keyfile1 pass1 alias1 www.test.com 730 US it@test.com
   TestCo MIS Massachusetts Boston
password: ***************
 confirm: ***************
Success!
```

cert-request

Purpose

Generates a certification request for a private key and its certificate. You specify the keyfile and entry within it that you want to generate for, as well as a file name to write the data to. You then send that file to the CA authority. You can use the cert-gen action to generate an entry in the keyfile for which you can request certification, or you can use a keyfile and alias that exist on the system. Use the cert-update action to update the self-signed certificate with the validated certificate. Enter the following:

- ***keyfile***—the name of the file containing the key and the self-signed certificate.
- ***password***—the password used to encrypt the private key when it was created.
- ***alias***—the entry within the certificate for which you are requesting a validation certificate.
- ***csrfile***—the file to which the certification request should be saved. The data is output in PEM format. You must send the contents of this file, along with your CSR, to the CA. By default, the file is written to the /cxc/certs directory.

When you invoke this action, you are required to enter and confirm your password. Enter the password that you used, in the cert-gen action, to create the key pair.

Syntax

```
cert-request keyFile password alias csrFile
```

Example

```
NNOS-E> cert-request keyfile1 pass1 alias1 CSRfile
password: ***************
 confirm: ***************
Success!
```

cert-update

Purpose

Loads a signed certificate onto AA-SBC and associates it with the key specified by alias. Use the action to update your self-signed certificate when the CA has returned a signed certificate. When you receive the file, you can put it anywhere on AA-SBC. However, if you store it in the directory /cxc/certs, it will be available to all boxes in the cluster. Enter the following:

- ***keyfile***—the name of the file containing the key and the self-signed certificate.
- ***password***—the password used to encrypt the private key when it was created.
- ***alias***—the entry within the certificate for which you are creating a validated certificate.
- ***certFile***—the signed certificate file returned from the CA.

You can use the cert-gen action to generate an entry in the keyfile for which you are requesting certification; generate the request using the cert-request action.

Syntax

```
cert-update keyFile [password] alias certFile
```

Example

```
NNOS-E> cert-update keyfile1 pass1 alias1 signed_1.cer
password: ***************
 confirm: ***************
Success!
```

clock

Purpose

Sets AA-SBC system time. The internal clock operates uses a 24-hour clock, beginning at midnight starting at 00:00 hours.

Syntax

```
clock hour:minutes
```

Actions

Example

The following example sets the system clock to 2:00 p.m.

```
NNOS-E> clock 14:00
Success!
NNOS-E> show clock

  time: 14:00:04 Mon 2006-01-30
uptime: 0 days 01:36:41
```

cls

Purpose

Clears the window in which your CLI session is active by moving your prompt and cursor to the first line of the display. Your terminal emulation program defines at what point the screen erasure begins (and therefore, which previous data is still displayed if you scroll backward).

Syntax

```
cls
```

Example

The following example clears the terminal screen and moves your cursor and prompt to the top line:

```
NNOS-E> cls

possible completions:
 clock
 cls
 cluster

NNOS-E> cls
```

collect

Purpose

The AA-SBC has the ability to collect support data and store it in a single compressed file to be downloaded and forwarded to the support team for analysis. A **collect** action has been created which allows you to collect the information necessary to troubleshoot problems occurring on the AA-SBC.

By default, the AA-SBC collects the following data when the **collect** action is executed.

- Configuration data, including the following:
  - Current running configuration (even if it has not been saved yet)
  - Current /cxc/cxc/cfg configuration file
  - Backup configuration files in /cxc/backup
  - Schema files (*.xsd in /cxc/web)
- Certificate files found in the /cxc/certs directory
- Status data which can be collected in two forms:
  - Text files that contain output equivalent to the status show commands
  - XML files that contain the same data, but in a structured format that is machine-readable and is used for automated analysis

  Status data can be collected in two different ways:

  - Default collection, in which a standard, pre-configured list of status classes is collected
  - Custom collection, in which status classes not included in the default list can be specified
- Crash files found in the /cxc_common/crash directory
- Log files found in the /cxc_common/log directory
- Directory contents

Collecting Data from a Cluster

By default, the collect action collects data only from the box on which it is executed. Cluster-wide data collection can be specified by adding the cluster parameter to the action.

To collect the default data throughout the cluster, you must specify the default parameter.

```
NNOS-E>collect default cluster
```

To collect custom data from a configured collect-group, specify the collect-group (in this example accounting is used).

```
NNOS-E>collect accounting cluster
```

When cluster-wide data collection is specified, each AA-SBC collects the appropriate data independently and simultaneously. The AA-SBC on which the **collect** action is executed then combines the resulting data into a single file.

Syntax

```
collect default
collect [group-name] [realm] [destination]file
```

Example

```
NNOS-E> collect default
Box 1:  collecting configuration data...
Box 1:  collected configuration data; Success!
Box 1:  collecting certificate files...
Box 1:  collected certificate files; Success!
Box 1:  collecting status data...
Box 1:  collected status data; Success!
Box 1:  collecting crash reports...
Box 1:  collected crash reports; Success!
Box 1:  collecting event logs...
Box 1:  collected event logs; Success!


Box 1:  finalizing to '/cxc_common/collect/collect.tar.gz'...
Box 1:  finalized; Success!

Elapsed time:  10.874 seconds
File size:     4.91M bytes
```

config

Purpose

Manipulates or saves the running configuration

- **merge**—merges the specified file into the current running configuration. If any properties overlap (set in both configuration files), the values from the file being merged in take precedence.
- **replace**—writes the specified file to the running configuration. All values are overwritten with the values of the new file.
- **save**—writes the running configuration to the default configuration file (/cxc/cxc.cfg), or if a pathname is supplied, to that file name. You can choose standard, verbose, or XML formats. Standard format only outputs properties with a value different from the default; verbose outputs every property.

  You can also save your configuration file in XML format. You can then import this XML file to other AA-SBC systems to create a saved configuration. This will save you time if you have identical configuration settings across systems in the cluster. With XML, you can also work on the configuration file offline. In the CLI, XML and standard/verbose CLI configuration files are interchangeable (functionally equivalent), and the default save location is the same.

  AA-SBC creates a numbered backup (cxc.cfg.#) with each execution, creating up to four backups. Files are saved in the /cxc/backup directory.

  This command works the same as the save global command.
- **setup**—creates a minimal operating configuration file from a setup script run from the top-level prompt. After answering a a series of questions at the command line, you can then PING the system to check connectivity.

Syntax

```
config merge fileName
config replace fileName
config save [standard | verbose | xml] [fileName]
config setup
```

Example

The following example saves the running configuration to XML format under a new file name.

```
NNOS-E> config merge /cxc/cxc.stock
Success!
NNOS-E> config replace /cxc/cxc.cfg
Success!
NNOS-E> config save xml cfg.xml
Success!
```

cpu-monitor

Purpose

Displays the percentages of CPU usage over time. The first column of the output indicates the system time stamp of the reading. The second column indicates CPU usage as a percentage of total CPU processing power. The third column, if present, indicates the number of seconds that the measurement has been unchanged. Use the **show cpu-usage** command to display usage levels at preset intervals over time.

In the first example, below, the system is using 0% of the total CPU. The value has been unchanged for 34 seconds. In example 2, the value is continuously changing. Example 3 shows the ASCII output of system under DOS attack with no DOS rules enabled to protect it. If you have a color ANSI terminal emulation program, the system will display CPU monitoring graphically, as shown in Example 4. Green bars show usage between 1% and 60%; yellow shows 61-80%, and red shows 81-100%.

Press ESC to cancel the display.

Syntax

```
cpu-monitor
```

Example

```
NNOS-E> cpu-monitor
Monitoring cpu usage; press Esc to cancel...
15:26:43 0% 34
```

Example

```
NNOS-E> cpu-monitor
Monitoring cpu usage; press Esc to cancel...
  21:23:23 9%
  21:23:24 7%
  21:23:25 7%
  21:23:26 6%
  21:23:27 6%
  21:23:28 9%
  21:23:29 7%
```

Example

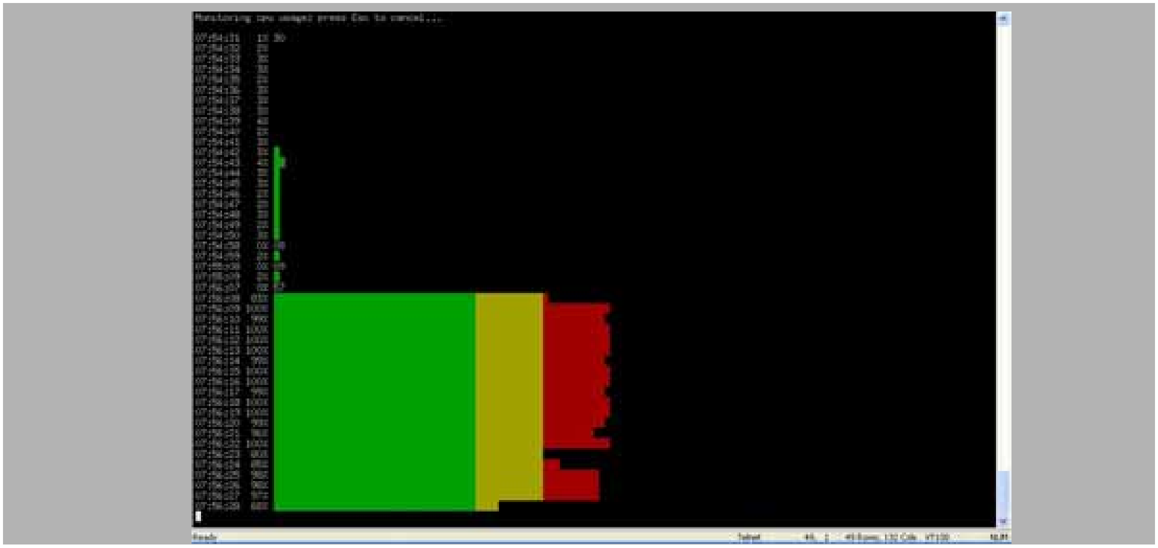
The following example shows first the ASCII output of a system under attack with no DOS rules enabled, and then the terminal emulation output of the same attack.

```
NNOS-E> cpu-monitor
Monitoring cpu usage; press Esc to cancel...
08:04:26 0% 13
08:04:27 34%
08:04:28 98%
08:04:29 99%
08:04:30 100%
08:04:32 100%
08:04:33 100%
08:04:34 100%
08:04:35 99%
08:04:36 100%
08:04:37 99%
08:04:38 99%
08:04:39 99%
08:04:40 77%
08:04:41 41%
08:04:42 43%
08:04:43 77%
08:04:44 55%
08:04:45 57%
08:04:46 33%
08:04:47 68%
08:04:48 39%
08:04:49 60%
08:04:50 54%
08:04:51 76%
08:04:52 56%
08:04:53 49%
08:04:54 32%
08:04:55 58%
08:04:56 51%
08:04:57 40%
```

Actions

```
08:04:58 29%
08:04:59 14%
08:05:00 12%
08:05:01 8%
08:05:02 6%
08:05:03 4%
08:05:04 2%
08:05:05 4%
08:05:06 3%
08:05:07 1% 01
08:05:08 3%
```

The following image is a terminal emulation output of a cpu-monitor action:



```
Monitoring cpu usage; press Esc to cancel...
```

csta-moc-commands

Purpose

Manages MOC clients, such as changing status for one, two, or all clients, or finding login status of clients.

- **update-moc**—changes the status of a MOC client to the status you specify by issuing a CSTA Call Update. Enter the From URI, in the form of a telephone number (e.g., tel:+14135551212 or the normalized 5551212), to change the status of the caller. Optionally, you can simultaneously change the "callee" to the same status by entering the To URI.
- **find-moc**—searches the list of MOC clients, listed in the CSTA and/or MOC caches, for one with the specified URI. Results of the action indicate whether or not the client is logged in.
- **reset-all-moc**—resets the status of all the MOC clients that are currently connected to AA-SBC. Status for all is changed to Available.

Syntax

```
csta-moc-commands update-moc {do-not-disturb | call-forward |
   call-connected | call-terminated | onhook | offhook} fromURI
   [toURI]
csta-moc-commands find-moc fromURI
csta-moc-commands reset-all-moc
```

Example

```
NNOS-E> csta-moc-commands find-moc tel:+9788235226
Device <tel:+9788235226> not found in CSTA cache.
Specified URI not found in MOC cache.
NNOS-E> csta-moc-commands find-moc 6474840
Device <6474840> found in CSTA cache.
```

csta-uri-normalization

Purpose

Tests the regular expression rules for URI normalization that are configured for a 3PCC server. This is a way to reference a configured server and run intended URI normalization rules against it to verify the applicability of the normalization properties. When you execute this action, you reference a server and indicate which type of normalization rules to test. AA-SBC then applies the rules in that configuration. For a full description of using URI normalization in third-party call control, see the 3pcc-servers object description. Note that issuing the action using the keyword **none** initiates no action.

To use this action, specify:

Actions

- a server type
- a reference to a server of that type
- the type of normalization for that server that you are testing
- the phone number to run the test against.

Syntax

```
csta-uri-normalization none
csta-uri-normalization {internal | broadworks | cisco | avaya |
   loopback} 3pccServerReference {incoming | outgoing | server}
   telNumber
```

Example

```
NNOS-E> csta-uri-normalization internal
   "vsp\enterprise\3pcc-servers\internal-csta-server "Internal
   Server"" incoming 5551212
```

database

Purpose

Deletes or cleans database records. This is for databases you configured with the master services' database object. You can clean or delete an entire database, or a specific table within the database. Use the **show database-tables** command to list the tables and their associated database.

Regular vacuuming is done automatically as part of the nightly maintenance and logs should show when a table is automatically vacuumed.

- **delete**—purges the database of entries contained in the specified database, or entries in the table within the database. The **database delete** action (without qualifiers) deletes all rows in all tables in the database.
- **vacuum**—based on the SQL VACUUM command, reclaims storage occupied by deleted entries and makes it available for re-use. The system can read and write to the table while the process is occurring.

- **vacuum-full**—based on SQL VACUUM command, reclaims storage occupied by deleted entries and makes it available for re-use. It also does more extensive processing, however, and as a result the table is not available for read/write operations during the process. To do a periodic “global” vacuum, use the **database vacuum-full system** command. If you receive a message telling you a specific table needs to be vacuumed, execute the **database vacuum-full system <table>** command.
- **drop**—deletes all data stored in the specified table and removes the table definition from the database schema.
- **repair**—Initiates database repair options. If you select the **data-recovery** option, the system recovers data that was removed by AA-SBC when it corrected a corrupted database. The **translate** option migrates earlier databases to a format compatible with release 3.2 and later.
- **initialize**—deletes all data and reinitializes the database.
- **snapshot**—Breaks the database into smaller pieces, each starting from either the beginning of the database or the last snapshot, and ending when this action is executed (either manually or as a task). This results in fewer and faster disk accesses and improved performance. Use this action to manually take a snapshot, or schedule periodic snapshots with the **task** object. (See Taking snapshots at regular intervals for more information.) You can access archived snapshots from the AA-SBC Management System **Call Logs** tab by clicking on the **Database Archives** link.

  Select a minimum number of records in the snapshot, either an integer or the **force** or **automatic** options. If you enter a number, AA-SBC takes the snapshot if there is at least one table that has at least that many records. Otherwise, it does not execute the action for that interval. The default number of records is three million. The **force** option takes the snapshot regardless of how many records there are in the database tables. The **automatic** option skips the snapshot if no table has 3 million records or more.

Taking snapshots at regular intervals

To take a snapshot at regular intervals, use the **task** object. You can, for example, schedule AA-SBC to take a snapshot every four hours. The content of the snapshot will depend on the settings of the snapshot option, but will contain the data that was written to the database from the end of the previous snapshot to four hours forward. Each snapshot will contain only those four hours worth of data. Queries can then be performed on snapshot segments instead of the whole database. When creating a database snapshot, AA-SBC:

1. takes the current timestamp and uses it as the part of the snapshot data directory name;
2. dumps the whole database up to the timestamp to a new database stored in the named data directory;
3. deletes from the running database all records with a timestamp up to the above set timestamp;
4. performs a vacuum analyze on the running database to reclaim disk space.

While a snapshot is in progress, all database reads and writes, as well as DOS queries, are performed as usual.

Syntax

```
database delete database [table]
database vacuum database [table]
database vacuum-full database [table]
database drop} database [table]
database repair {translate | data-recovery}
database initialize [database-path]
database snapshot database {integer | force | automatic}
```

Example

```
NNOS-E> database vacuum status cpuusage
Are you sure (y or n)? y
Success!
NNOS-E>
```

database-backup

Purpose

Executes a database backup or restore operation. A backup saves the database to the path you specify. The restore actions loads the specified database file from the location you specify to AA-SBC. Any restore action adds entries from that file to the database. (If your goal is to overwrite the database, then you should first use the **database delete** action and then use the **database-backup restore** action.)

When you supply a path name, you are also giving a name to the database file. AA-SBC saves the file to /cxc/pg_dump/*name*. Do not specify a path name unless it begins with /cxc/pg_dump/. For example, if you specify db1, AA-SBC saves it to / cxc/pg_dump/db1. Or, you could specify, /cxc/pg_dump/db1. However, if you specify /cxc/db1, the operation will fail.

Note that by default AA-SBC uses BZIP2 compression. This format is optimized for size, but can take longer to produce. If you would prefer to use GZIP compression, which is faster but results in a 30-40% larger archive, you can do so by supplying the **gz** suffix when you initiate the action. For example:

Enter this file name at the command line...	Get an archive of this type...
DBbackup	DBbackup.bz2
DBbackup.gz	DBbackup.gz

Syntax

```
database-backup {backup | restore} {log | system | status dos |
   directory | accounting} databasePath
```

Example

```
NNOS-E> database-backup restore system /cxc/pg_dump/sysDB
Are you sure (y or n)? y
Starting database restore as a background operation.
 -- this may take a very long time --
Please check database-maintenance-status for notification when this
   operation is complete.
NNOS-E> show database-maintenance-status
```

Actions

```
maintenance-status: idle
```

directory-reset

Purpose

Resets the enterprise directory, causing AA-SBC to reread the directory and update the user information. Use this action when you have added users and want AA-SBC to retrieve the new entries.

Enter the name of the VSP that houses the directory. In addition, you can set a directory purge action of true or false:

- **true**—clears out the contents of the database and then repopulates it.
- **false**—updates the database but leaves users that are no longer in the directory itself in the database.

If you do not enter a VSP name, the system uses the VSP **default**. For the directory-reset action, the default purge action is **true**.

You can cancel this action when it is in progress using the directory-reset-cancel action. You can also schedule this action as part of routine maintenance using the **task** object.

Syntax

```
directory-reset [vsp] [true | false]
```

Example

```
NNOS-E> directory-reset
Success!
NNOS-E> directory-reset vsp2 false
The specified vsp was not found
NNOS-E>
```

directory-reset-cancel

Purpose

Cancels the execution of an in-progress directory-reset action. When the **directory-reset-cancel** action is invoked, it immediately stops the reset action—some entries are updated and some are not. Use this action with caution as it leaves the directory entries in a mixed state.

Syntax

```
directory-reset-cancel
```

Example

```
NNOS-E> directory-reset-cancel
directory-reset-cancel has been submitted.  Please check
   directory-status for progress.
NNOS-E>
```

disconnect-call

Purpose

Disconnects the specified call based on the session ID. If a call hasn't yet been answered, this action sends a “403 Forbidden” response to the UA that initiated the call, and a CANCEL request to the called UA. If the call has already been cancelled, the action sends BYE requests to both UAs (informing all involved parties that the call has been shut down). Use the **show active-call** command to retrieve the ID. You can also do this from the Call Logs **Session** pages in the AA-SBC Management System. From the GUI, you can list the active calls and then select and disconnect. Use the terminate-call action if the endpoints are no longer reachable.

Syntax

```
disconnect-call sessionID
```

Example

```
NNOS-E> disconnect-call 0x4c22932e1cf4ba6
Success!
NNOS-E>
```

display

Purpose

Temporarily changes the display output without changing your configuration. After ending your CLI session, AA-SBC erases the changes and uses the settings in your configuration at the next session. Select either:

- **paged**—the CLI outputs text a page at a time, pausing the display with the **--More--** prompt. You set the number of lines displayed before the prompt. The prompt accepts the following keystrokes:

Keystroke	Result
ENTER	Outputs the next line of text.
TAB	Outputs the remainder of the text.
ESC or Q or q	Cancels the display, outputs no more text, and returns to the prompt.
any other keystroke	Outputs the next page of text.

- **scrolled**—the CLI scrolls text continuously.

To set the output display in your configuration, use the cli object.

Syntax

```
display {paged numberOfLines | scrolled}
```

Example

```
NNOS-E> display paged 12
NNOS-E> ?
 archive                     Run the archiving task for a given vsp
 arp-delete                  flush the ARP cache
 autonomous-ip-lookup        Actions to determine Autonomous IP group
 call-lookup                 Actions for Dial Plan operations
 clock                       set the system time
 cls                         clear terminal screen
 config                      configuration commands
 database                    delete database records or vacuum tables
```

```
--More--
```

dns

Purpose

Executes a variety of DNS actions. Use the **dns** object to configure how AA-SBC services DNS requests. Select one of the following options:

- **lookup**—executes a DNS lookup on a named host. You can specify a server to query; otherwise AA-SBC uses the server(s) configured through the resolver server. You can set a timeout for the request, either a number of milliseconds or that the request never timeout (forever). Optionally you can set the record type that AA-SBC returns:
  - **A**—an IP address (the default).
  - **PTR**—pointer record, mapping IP address to the canonical name.
  - **SRV**—service location record.
  - **NAPTR**—name authority pointers, mapping a domain name to general information such as a URI.
  - **CNAME**—canonical name, mapping any name aliases.
  - **NS**—name server record, mapping a domain name to authoritative DNS servers for that name.
- **flush-cache**—Flushes all dynamic entries from the DNS resolver cache of all processes. Optionally, you can specify an individual process cache to flush.
- **reset-servers**—Flushes all dynamic entries from the server DNS resolver cache and resets the DNS sockets. It is possible for a DNS socket has become nonfunctional but AA-SBC continues sending DNS messages to it. This action closes and then reopens all DNS sockets. In addition it resets server counts, refreshes the server, and sets the administrative state to UP. By default, the action impacts all servers configured as resolver servers. Optionally, you can specify an individual server.

- **delete-entry**—Deletes the specified record from the DNS cache. Enter the name, and, optionally, the record type of the entry to be deleted. You can delete the entry from all configured servers (the default), or a specific server. Use the **show dns-cache** command to display names of the entries in the cache. a server.

Syntax

```
dns lookup hostName [A | PTR | SRV | NAPTR | CNAME | NS] [serverName]
   [forever | milliseconds]
dns flush-cache [process]
dns reset-servers [serverName]
dns delete-entry name [A | PTR |SRV | NAPTR | CNAME | NS] [serverName]
```

Example

```
NNOS-E> dns lookup www.yahoo.com
www.yahoo.com      IN
     A tag-mine Resolved 60 69.147.76.15
```

dos-delete-rules

Purpose

Deletes all or a specific configured denial-of-service (DOS) rules. Use the **show dos-rule** command to see the current rules. Rules can be deleted on the local system or for the cluster. By default the action deletes all rules across the cluster.

AA-SBC automatically creates DOS rules as a result of the configured policies. The policy selects which table rows to consider, the threshold for instances, and the frequency (period) of comparison. When the threshold is reached for a given period, AA-SBC generates a rule in the kernel to block traffic meeting that selection criteria. The rule will persist as long as any traffic matching the rule is seen within the user-configurable inactivity-timeout period. AA-SBC automatically deletes a rule when the inactivity timer expires,. You can delete rules manually using this action.

See Chapter 19, “Denial of Service (DOS) objects” for more information.

Syntax

```
dos-delete-rules [cluster | local] [all | number]
```

Example

```
NNOS-E> dos-delete-rules
Success!
```

enum-lookup

Purpose

Enter a phone number to resolve. As a resolver, AA-SBC obtains resource records from servers on behalf of resident or requesting applications. In addition, it maps a server to a domain name.

Syntax

```
enum-lookup phoneNumber [domainName] [server]
```

Example

```
NNOS-E> config vsp enum mapping 15085551212
Creating 'mapping 15085551212'
config mapping 15085551212> set url SIP:skd@skdPC:5060;transport=TLS
config mapping 15085551212> exit
Do you want to commit your changes before you exit (y or n)? y
Do you want to update the startup configuration (y or n)? y
NNOS-E> enum-lookup 15085551212
Phone 15085551212 has a mapping to URL
   SIP:skd@skdPC:5060;transport=TLS

NNOS-E>
```

ethernet-negotiate

Purpose

Causes the specified Ethernet interface to renegotiate its parameters. To see the current settings, execute the **show ethernet** command. Enter the Ethernet interface you want to effect, eth0 through eth19.

Syntax

```
ethernet-negotiate ethX
```

Actions

Example

```
NNOS-E> ethernet-negotiate eth0
Success!
NNOS-E> show ethernet

name  link speed duplex autoneg
----  ---- ----- ------ -------
eth0  up   1Gb   full   enabled
eth1  down              enabled
eth2  down              enabled
eth3  down              enabled
```

expression

Purpose

Provides actions to test match and replacement in regular expressions. A scan option provides line-by-line processing output to help with debugging. Remember that you must use quotation marks around special characters at the command line. Select either:

- **match**—tests a URI, allowing you to determine whether an expression you created matches a supplied string.
- **replace**—tests both the regular expression and the replacement for a URI against the supplied string. Supply both and the system responds with the resulting output. AA-SBC executes the replacement on the first occurrence of the expression. To replace all instances, use the **all** keyword option.
- **scan**—debugs a URI and regular expression by providing pass-by-pass information on how the system is processing the string.

Syntax

```
expression match string regExp
expression replace string regExp replacement [all]
expression scan string regExp
```

Example

```
NNOS-E> expression match 5551212 ^([0-9]{7})$
Success!
NNOS-E> expression replace 5551212 ^([0-9]{7})$ "tel:+1413\1"
tel:+14135551212
```

```
NNOS-E> expression scan 5551212  ^([0-9]{7})$
match at offset 0, length 7
...substring 0: '5551212'
...substring 1: '5551212'
```

external-normalization

Purpose

Manages the file used to maintain DNIS-to-ANI translation data.These mappings are checked prior to any dial-plan lookups. Typically, this action is initiated though the task object scheduler. Select one of the following operations:

- **replace-file**—Pulls data from the named file into the external normalization database of the SIP process. All previous mappings are removed. (The mappings known to AA-SBC can be displayed using the **show external-normalization** status provider.) Enter a file path or browse the system for a file name. The default normalization file is **/cxc/normalization.xml**.
- **replace-url**—Identifies the external service that hosts the normalization (mapping) data. Specify a host name or IP address.
- **flush**—removes all entries from the internal normalization cache. (It does not affect the normalization.xml file.)

You can also schedule this action as part of routine maintenance using the task object.

Syntax

```
external-normalization {replace-file fileName | replace-url source |
   flush}
```

Example

```
NNOS-E> external-normalization flush
Success!
NNOS-E> show external-normalization
match        pri  hits  ALT-REQ-USR  ALT-TO-USR  ALT-FROM-USR  DLG  tag
-----        ---  ----  -----------  ----------  ------------  ---  ---

NNOS-E> external-normalization replace-file normalization.xml
Success!

NNOS-E> show external-normalization
match        pri  hits  ALT-REQ-USR  ALT-TO-USR  ALT-FROM-USR  DLG  tag
```

```
-----       ---  ----  -----------  ----------  ------------  ---  ---
899101201   99   0                            **12027423317   OUT
1202742331  99   0    **8991012017                            IN   abc
NNOS-E> external-normalization replace-url ftp://myserver.foo.com/
   external-normalization.xml
Success!
```

external-presence

Purpose

Clears all or a specified entry from the external presence cache. The external cache is the database running on the backup system in a cluster configuration. The primary or master appliance contains the presence cache from which the external presence cache is mirrored. If a failover happens, the external cache becomes the master cache. See the presence action for information on managing the master cache.

Select one of the following operations:

- **delete**—deletes the specified entry from the external presence cache. Enter the URL for the entry, which can be found in the **show presence-cache-external** command.
- **flush**—removes all entries from the external presence cache.

You can also schedule this action as part of routine maintenance using the task object.

Syntax

```
external-presence {delete url | flush}
```

Example

```
NNOS-E> show presence-cache-external
url: 2078548355@elmaple.com
Box: 0.0.0.0
state: Online
prestype: Voice
LastRegisteredTime: 0
ExpireInterval: 4294967295
numCurrSubscribers: 1
url: 2078548357@elmaple.com
Box: 0.0.0.0
state: Online
prestype: Voice
```

```
LastRegisteredTime: 0
ExpireInterval: 4294967295
numCurrSubscribers: 3
NNOS-E> external-presence flush
Success!
```

external-session

Purpose

Removes all entries from the external CSTA SIP session cache. If configured to do so, the cache contains state information for all active CSTA sessions on all boxes in the cluster. This action is typically used to clear the cache without rebooting in the event of a failover recovery. Use the install examine action to clear the cache on an individual box.

To ensure that boxes share this information, set **share-signaling-entries** to **true** in the cluster object.

You can also schedule this action as part of routine maintenance using the task object.

Syntax

```
external-session flush
```

Example

```
NNOS-E> external-session flush
Success!
```

file

Purpose

Performs file management operations. The file actions support a number of protocols (HTTP, HTTPS, FTP, etc.) and can be used to manage a variety of file types, for example, configuration files, certificates, patches, or licenses. All operations begin with \cxc as the root directory. Select one of the following operations:

- **erase**—Deletes the specified file.

- **purge**—deletes one or more files. Use a wildcard to specify more than one (e.g., /cxc_common/announcements/*). You can select a minimum age for the file(s)—the system deletes anything older. See Setting time and time intervals for information on entry format requirements for minimum age.
- **fetch**—Moves a file from a specified location to AA-SBC.
- **send**—Copies a file from AA-SBC to the specified location.

Syntax

```
file erase fileName
file purge fileName [age]
file fetch sourceURL [destinationFile]
file send sourceFile [destinationURL]
```

Example

```
NNOS-E> file erase signature.txt
Success!
NNOS-E> file fetch https://192.168.10.10/cxc.cfg
Success!
```

file-based-word-lists-refresh

Purpose

Rereads any saved word-list or url-list **file** entry into memory. Use this if you have made a change to the file and want the new words, expressions, or domains read into memory.

You can also schedule this action as part of routine maintenance using the task object.

Syntax

```
file-based-word-lists-refresh
```

Example

```
NNOS-E> file-based-word-lists-refresh
Success!
```

file-mirror-service

Purpose

Manages mirrored files and the file database. To enable file mirroring, use the file-mirror master service. File mirroring sets all participating AA-SBC devices to share particular files, such as media recordings, log files, etc., making them highly available. Using this action, you can execute the following operations:

- **make-available**—moves the named file from its current location to the common directory of the first highly available directory. This is the first entry in the file-mirror **file-mirror-directory** property configuration. The master then distributes the file across the cluster. Use this option to mimic the file mirror function for files that are not automatically managed under the service.
- **fetch**—validates whether the specified file is up-to-date on the disk. If it is not, the action retrieves the current file from the master.
- **delete**—removes the specified file form both the local disk and the shared database. The action also sends a message to the master, instructing it to remove the file from each backup box.
- **find**—returns a list of full path name matches to the relative path name that you enter.

All operations take a file name.

Syntax

```
file-mirror-service {make-available | fetch | delete | find} filePath
```

Example

```
NNOS-E> show file-mirror-db-record

canonical-file-name   timestamp   file-permission   last-update-from-b
   ox-number
-------------------   ---------   ---------------   ------------------
   ---------
/cxc_common/mirror1/file1.txt 15:24:16 Tue 2007-10-16  33188
   1
/cxc_common/mirror1/file3.txt 15:24:48 Tue 2007-10-16  33188
   1
/cxc_common/mirror2/file2.txt 15:26:21 Tue 2007-10-16  33188
   1
```

Actions

```
/cxc_common/mirror2/file3.txt 15:26:14 Tue 2007-10-16  33188
   1

NNOS-E> file-mirror-service find file1.txt
Found file /cxc_common/mirror1/file1.txt

NNOS-E> file-mirror-service find file3.txt
Found file /cxc_common/mirror1/file3.txt
Found file /cxc_common/mirror2/file3.txt
```

file-play

Purpose

Places a call to the specified SIP URI, plays the .WAV file, and then disconnects the call. This could be a .WAV file you recorded and moved to AA-SBC, for instance with the **file fetch** action.

Compare this to the playback action. The playback action plays recorded sessions only (AA-SBC takes care of mixing the media for playing). This action plays any audio file. For example, if you made a file using the mix-session action, you can play it using **file-play**.

Enter the following information:

- ***filename***—the location of the .WAV file you want played.
- ***to***—the SIP URI that specifies where to place the call to.
- ***from***—a SIP URI that appears as the caller ID.
- ***transport***—the transport protocol to use, either any, UDP, TCP, or TLS.

Syntax

```
file-play fileName to [from] [transport]
```

Example

```
NNOS-E> file-play greeting.wav sip:users@cov.com
   sip:management@cov.com
Success!
```

file-play-verify

Purpose

Verifies that a WAV file is of a format supported by AA-SBC (for example, PCM16/PCMA/PCMU, 16000/8000 sample rate, single channel, and others). Use this action to validate a file used for playout or for insertion as an introduction (see the media object) or periodic-announcement.

Syntax

```
file-play-verify fileName
```

Example

```
NNOS-E> file-play-verify /cxc/recordings/intro1
Success!
```

file-transfer-delete

Purpose

Deletes a file that was added to AA-SBC via a file transfer (e.g., via IMs) and entered in the database. (The database maintains a record of all files transferred.) Deleting the file removes the entry from the database.

Enter the following:

- ***sessionID***—To find the session ID, use the **Call Logs** tab in the AA-SBC Management System or the **show file-transfer-files** command in the CLI.
- ***fileName***—the name of the file to delete.
- ***identifier***—the unique identifier that distinguishes the file in the event that the same named file was transferred more than once during a single session.

Syntax

```
file-transfer-delete sessionID fileName identifier
```

Example

```
NNOS-E> file-transfer-delete ca3e3764f07c42f5b7b6eda269d2d0c4
   greeting.wav 43ab
Success!
```

file-transfer-delete-old

Purpose

Invokes AA-SBC to delete all files added to AA-SBC via a file transfer that are older than the specified number of days or seconds. Enter a number and a unit of measure. By default, AA-SBC deletes transferred files that are older than seven days when you execute this command.

You can also schedule this action as part of routine maintenance using the **task** object.

Syntax

```
file-transfer-delete-old age [days | seconds]
```

Example

```
NNOS-E> file-transfer-delete-old 30
Success!
```

file-transfer-retrieve

Purpose

Creates a link to the specified file transfer so that the AA-SBC Management System can retrieve the file. This action is primarily for use with the GUI only.

Enter the following:

- ***sessionID***—To find the session ID, use the **Call Logs** tab in the AA-SBC Management System or the **show file-transfer-files** command in the CLI.
- ***fileName***—the name of the file to retrieve.
- ***identifier***—the unique identifier that distinguishes the file in the event that the same named file was transferred more than once during a single session.

- ***file***—the path to the link created for the file.

Syntax

```
file-transfer-delete sessionID fileName identifier file
```

Example

```
NNOS-E> file-transfer-retrieve ca3e3764f07c42f5b7b6eda269d2d0c4
   greeting.wav 43ab /cxc/web/tmp928421
Success!
```

format

Purpose

Formats the specified system hard drive. Enter the drive you wish to format and the file system to use on that drive. Use this action with caution, as it formats or reformats the specified drive, removing all existing data. Note that the format action also automatically performs the equivalent of the add-device and mount actions. If you have data on a drive that you want to maintain, be sure to manually execute these two actions.

Syntax

```
format {usb | data-1 | data-2} [reiser-3 | xfs | vfat]
```

Example

```
NNOS-E> format data-1
Are you sure (y or n)? y

Success!
```

ftrace

Purpose

Manages debug tracing results that are directed to a file. You must have an existing settings file to start writing trace results to a file. The settings file defines the trace class and severity to record. You can create a settings file with the AA-SBC CLI **trace** command or another program.

Select one of the following options

- **start**—begins writing tracing results to a file. Specify the process that you want to collect debugging information on, and a file name to write the information to. Optionally, you can specify a settings file. The default settings file used by AA-SBC, if you do not specify a settings file name, is **target.ini**. However, you must create that file before you can use this action.
- **stop**—ends the tracing action for a specific process to a specific file.
- **remove**—deletes the specified file for a given target.
- **import**—imports a trace file into the log database. Specify the full path name of the file to import. The default table to import the file to is **TraceStruct**.

Syntax

```
ftrace start process target [settings]
ftrace stop process target
ftrace remove process target
ftrace import file [table]
```

Example

```
NNOS-E> ftrace start sip /trace/sip
Success!

NNOS-E> ftrace stop monitor /trace/mtr
Success!
```

group-down

Purpose

Simulates a VRRP group becoming non-operational. You can use this action to test your configuration. If configured for failover, when you execute this action, AA-SBC will fail over interfaces and master services to the backup box. If you execute this action without such a configuration, AA-SBC will bring down and then restore the specified group.

Syntax

```
group-down vrrpGroup
```

Example

```
NNOS-E> group-down 1
Success!
```

h323-reregister-gatekeeper

Purpose

Sends an UNREGISTER and then REGISTER request to an external gatekeeper, which renegotiates the registration. Use the s**how h323-external-gatekeeper** command to list gatekeepers and their IP addresses.

Syntax

```
h323-reregister-gatekeeper remoteIP
```

Example

```
NNOS-E> show h323-external-gatekeepers
       remote-address: 172.10.100.10:1719
        local-address: 172.30.0.143:1719
             regstate: registered
 last-registered-time: 06:51:26 Mon 2008-10-06
registration-interval: 3600
        gatekeeper-id: OpenH323GK
          endpoint-id: 1290_endp
          activecalls: 0
           default-gw: false
```

Actions

```
NNOS-E> h323-reregister-gatekeeper 172.10.100.10
Success!
```

h323-unregister-gatekeeper

Purpose

Sends an UNREGISTER request to an external gatekeeper. Use the s**how h323-external-gatekeeper** command to list gatekeepers and their IP addresses.

Syntax

```
h323-unregister-gatekeeper remoteIP
```

Example

```
NNOS-E> h323-unregister-gatekeeper 172.10.100.10
Success!
```

install

Purpose

Manages system software releases and network interface cards (NICs). Note that if you have a file on your PC, and want to install it, you can either use PSCP (if you have SSH configured on AA-SBC) or TFTP (if you have TFTP configured on AA-SBC). Or, you can use the AA-SBC Management System **Update Software** tool.

Select one of the following:

- **file**—installs a file that exists on AA-SBC in a format that is install-ready (e.g., decompressed). Enter the file name. Optionally, specify whether to install the file on the local box or the entire cluster. However, to upgrade members in a cluster without interrupting call flow, use the **controlled** option. By default, AA-SBC installs the file on the device.

- **url**—first downloads the specified file, and then installs it onto AA-SBC. Enter a URL for the software upgrade. Optionally, specify whether to install the file on the local box or the entire cluster. However, to upgrade members in a cluster without interrupting call flow, use the **controlled** option. By default, AA-SBC installs the file on the device.
- **nic**—Updates the stored MAC addresses and associates them to interface names, making the system aware of interfaces it can use. Execute this action if you purchased a new NIC after your system was up and running (i.e., you are adding a NIC). Optionally, for a faster installation, you can specify the hardware platform instead of waiting for AA-SBC to read the IPMI.
- **nic-reinitialize**—Wipes the information from a NIC. Use this action if you selected the wrong system model with the **nic** action, but only if instructed to do so by technical support.
- **module**—Forces installation of an older module from a release marked as good. This action is used to help regain network connectivity should there be problems. You can type a question mark at the command line to see the available modules (**install module ?**).
- **cancel**—Cancels the operations initiated by the install file controlled action.

You can also schedule this action as part of routine maintenance using the **task** object.

Syntax

```
install file source [box | cluster | controlled]
install url source [box | cluster | controlled]
install nic [model]
install nic-reinitialize
install module kernelModule
install cancel
```

Example

```
NNOS-E> install file release.tar.gz controlled
Are you sure (y or n)? y
sending release.tar.gz to 172.66.0.10...
sending release.tar.gz to 172.66.0.11...
sending release.tar.gz to 172.66.0.12...
Restarting 172.66.0.10...
Restarting 172.66.0.11...
Restarting 172.66.0.12...
NNOS-E is restarting...
NNOS-E>
```

install examine

Purpose

Displays information about files used for upgrade. Typically this information indicates build, platform, prerequisite, and component data.

Syntax

```
install examine sourcePath
```

Example

```
NNOS-E> install examine /releases/release-b3.4-32250.tar.gz
----------------------------------
/releases/release-b3.4-32250.tar.gz
----------------------------------
summary NNOS-E Application
description
type simple
restart warm
variety app
platform all
platformType none
version 3.4.99
build 32250
branch b3.4
prereqOSBranch
prereqOSBuild 0
prereqAppBranch
prereqAppBuild 0
name
within
untarDir /cxc
preInstallScript
installScript install/install.sh
InstallComponents
 component[1] os 1.6 "" 32170
InstallComponents
 component[1] postgres "07.02.0005 PostgreSQL 8.1.2"
 component[2] postgres "07.02.0005 PostgreSQL 8.1.9"
InstallComponents
 component[1] kernel 2.6.11-6-cov kernel-2.6.11-6
```

internal-session

Purpose

Removes all entries from the internal CSTA SIP session cache. The cache contains state information for all active CSTA sessions being handled by the local AA-SBC devices. This action is typically used to clear the cache without rebooting in the event of a failover recovery. Use the external-session action to clear state data for all boxes in the cluster.

Syntax

```
internal-session flush
```

Example

```
NNOS-E> internal-session flush
Success!
```

jtapi-control

Purpose

Manages terminals associated with 3PCC servers and tests PBX connections. See the csta-settings object for a general description, and Identifying the active device for a discussion of how AA-SBC uses device ID information. See the Cisco online documentation, *Partitions and Calling Search Spaces*, for complete information on Cisco partitions.

- **get-terminals**—returns the unique identifier associated with the terminal(s) residing at the specified phone address. Use the output of this action with the **set-active-terminal** action.
- **set-active-terminal**—sets the active terminal for a phone address. Specify the terminal ID and the address to indicate which terminal is active for that address. Use the **get-terminals** option to returns a list of IDs.
- **clear-active-terminal**—removes the active terminal designation from the currently active terminal for a specific phone address. You can use **set-active-terminal** to overwrite the selection; use this to remove without replacing the active terminal selection.

Actions

- **select-active-terminal**—clears the current active terminal designation and then rings all terminals associated with the specified address. The terminal used to answer the query becomes the active terminal.
- **get-all-terminals**—returns, listed by phone address, all terminals for all users currently logged in to MOC.
- **test-pbx-connection**—used for debugging, tests to make sure that all terminals have a connection to the PBX. Specify a phone address, and AA-SBC tests each configured 3PCC server to see if that number can reach the PBX through the server.
- **reload-terminals-from-termdb**—refreshes the AA-SBC list of active terminals by overwriting the current list with the list stored in the database.
- **set-default-partition**—sets the partition that the specified address uses when involved in call operations. Enter the partition name and the phone address. The default partition can also be set with the **default-partition** property of the session configuration csta-settings object, but the setting of this action overrides the configuration.
- **set-terminal-partition**—sets the partition that a specific terminal within a specified address uses when involved in call operations. Enter the terminal ID (use the **get-terminals** option to returns a list of IDs) partition name and the phone address.
- **show-connections**—lists all JTAPI connections and their status.

Syntax

```
jtapi-control get-terminals phoneAddress
jtapi-control set-active-terminal terminalID phoneAddress
jtapi-control clear-active-terminal phoneAddress
jtapi-control select-active-terminal phoneAddress
jtapi-control get-all-terminals
jtapi-control test-pbx-connection phoneAddress
jtapi-control reload-terminals-from-termdb
jtapi-control set-default-partition partition phoneAddress
jtapi-control set-terminal-partition terminal partition phoneAddress
jtapi-control show-connections
```

Example

```
NNOS-E> jtapi-control get-all-terminals

# Provider
# Monitored Addresses 3
```

```
Provider: 172.30.3.201 Address: 54815 Active Terminal:
Provider: 172.30.3.201 Address: 54804 Active Terminal: SEP003094C3D233
Provider: 172.30.3.201 Address: 54810 Active Terminal:
# Provider
# Monitored Addresses 1
Provider: 172.30.3.199 Address: 77704 Active Terminal:

NNOS-E> jtapi-control get-terminals 77702
SEP000E0C774E0B

NNOS-E> jtapi-control test-pbx-connection 77702
Cisco Call Manager 6.0
 Provider: 172.30.3.199 16
  Address=77702 Exists
   Terminal=SEP000E0C774E0B

cisco 4.0
 Provider: 172.30.3.201 16
  Address=77702 Not found on Provider
```

lcr

Purpose

Provides utilities for route server. This action is only available from the CLI if the route-server master service is enabled. Select one of the following actions:

- **replace-file**—replaces the existing route server routing table with the contents of the specified file. Use this to update the routing definition database on the route server. Enter the name of a properly formatted XML file; the default file name is /cxc/carrier_routing.xml. You cannot specify either the current or the most recent backup routing file. (To replace the routing table with the most recent backup, use the **revert** option.)
- **commit**—writes appended route server routing table entries from memory to the routing table, clearing the memory. (The memory provides a temporary backup holding area.)
- **replace-url**—replaces the existing route server routing table with the contents of the file found at the specified URL. Use this to update the routing definition database on the route server. Enter the name of a properly formatted XML file; the default file name is /cxc/carrier_routing.xml.
- **flush**—removes all entries from the route server routing table on the master.

Actions

- **revert**—reverts the existing route server routing table to the table in use prior to the last update.
- **lookup**—tests retrieval from the route server routing table. Results display all routes that match the specified To, and optionally the From, URI(s).

Syntax

```
lcr replace-file fileName
lcr commit
lcr replace-url urlSource
lcr flush
lcr revert
lcr lookup toURL [fromURL]
```

Example

```
NNOS-E> lcr lookup 9788972990@acmepacket.com 7818972990@company.com

-----------------------------------------------------------------
Carrier                      Endpoint                    Mapping
-----------------------------------------------------------------
S - 10pct Mup Customer,    gateway1        ANI:7819376550

S - 10pct Mup Customer,    gateway10       ANI:7819376550

N - 10pct Mup Customer,    gateway4
-----------------------------------------------------------------
Total routes: 3
```

license

Purpose

Manages the licensed software that contains the your specific product features. You must install a license to unlock your customer-specific features. You can fetch a license from the AA-SBC license server, or apply a license that is already on your local system. Use the **show licenses** command to see whether a license is installed or to determine the license name.

A license contains one or more features. You can have multiple licenses active at any time (displayed with **show licenses**). You can apply, refresh, and revoke licenses, but not the individual features within a license.

- **fetch**—downloads a license from the AA-SBC license server and installs it on your system. Before executing this action be sure that you have a connection to the public Internet and that port 616 is available and is not blocked by a firewall. You must supply a key to access your file; copy and paste the key you received from your sales representative. Optionally, you can specify a different server. The default server path is https://license.covergence.com:616/.
- **apply**—applies the specified license file, which must be stored on the system from which the action is executed. Use this if you have multiple devices as part of your license agreement and you have already saved the file locally. Optionally you can specify whether the license application is temporary (the life of the current session) or permanent. By default it is temporary.
- **revoke**—disables the specified license. You might use this to disable a feature for testing, for example. Optionally you can specify whether the change is temporary (the life of the current session) or permanent. By default it is temporary.
- **refresh**—Refreshes a license expiration or feature change. Use this action if you have an existing license nearing expiration or an evaluation copy. When you execute the action, AA-SBC logs in to the license server with the key stored key and updates the license (if available) with a later expiration time. Optionally, you can specify a different server. The default server path is https://license.covergence.com:616/.

Syntax

```
license fetch key [server]
license apply file [temporary | permanent]
license revoke license [temporary | permanent]
license refresh license [server]
```

Example

```
NNOS-E> show licenses

     name: ABCco LICENSE
description: LICENSE for company ABC
        key: 87702f9a-be13-9974-83904-d00b7e4ab51f
    expires: 12.31.06
       file: license.xml

NNOS-E> license refresh ABCco License
```

```
Success!
```

linksys

Purpose

Manages the Linksys certificate process. Linksys equipment supports a proprietary version of SRTP. It uses SIP INFO messages to exchange credentials (in mini-certificates) and securely distribute the key used to encrypt/decrypt the RTP packets. The RTP encryption is a variation of *RFC 3711, The Secure Real-time Transport Protocol (SRTP)*. Linksys uses the same encryption algorithm (AES-CM-128), but uses HMAC-MD5 instead of HMAC-SHA1 for authentication.

AA-SBC must have access to the key used to generate the mini-certificates for participation in encryption/decryption. This action can generate the mini-certificate and private key needed by each phone.

Note: You must have a root certificate loaded on the system for this action to be successful. The default location for the root certificate is **/cxc/certs/linksys_ca.pem**.

The linksys action provides five tools:

- **mini-certificate**—Creates a mini-certificate, which will later be used by a Linksys phone to exchange an encrypted symmetric key. When both phones in a call support cryptographic exchange, use this action to create a mini-certificate that is sent in an INFO message to the other phone. (You must execute this action for both phones.) After exchanging mini-certificate, the phones can then exchange an encrypted symmetric key.

  Enter the following fields to generate a mini-certificate:

  - ***userID***—a name that identifies this phone (subscriber) to the other party. The user ID can be up to 32 characters.
  - ***displayName***—a name used by the caller to verify that the callee is the intended call recipient. Enter the user ID field in the Request URI of the INVITE message sent to the proxy server by the caller UAC when making a call to this subscriber (UAS). The display name can be up to 16 characters.

  - ***expiration***—the date and time at which this mini-certificate expires. Enter the date in the format *hh*:*mm*:*ssyyyy-mm-dd*.
  - ***filename***—a name for an output file that will contain the mini-certificate and private key. If you do not specify a file name, the output is not written to a file.

Once you execute this option, AA-SBC returns the content of the mini-certificate and the SRTP private key. You can copy and paste each of these fields into your Web GUI for your phone (or other software interface), as well as test the certificate using the **check-mini-cert** option.

- **generate-ca-key**—generates a Linksys/Sipura CA key. This is the public/private key pair that acts as the Sipura certificate authority. It is needed to generate the mini-certificates for each phone and during the key exchange.

  The key is stored in **/cxc/certs/linksys_ca.pem**. When executing this action, you can specify whether to overwrite any previous CA key. The default setting, **false**, does not overwrite the key. Set the field to **true** to force an overwrite.
- **check-mini-cert**—verifies the contents of a certificate created with the **mini-certificate** option. When executed, AA-SBC checks the expiration date and signature of the certificate. Enter the content of the mini-certificate to invoke this option.
- **display-mini-cert**—displays the values of the fields that you entered when you created the mini-certificate. Paste the encoded certificate into the system to display output in a readable format.
- **display-msg**—displays the fields of the SIP INFO messages that were created during key exchange. After a call is established, endpoints send INFO messages to negotiate key exchange. The bodies of these messages are base-64 encoded. Enter a message (which can be taken from the call log or a trace) to display the message in a readable format.

Syntax

```
linksys mini-certificate userID displayName expiration [fileName]
linksys generate-ca-key [false | true]
linksys check-mini-cert certContent
linksys display-mini-cert certContent
linksys display-msg message
```

Example

```
NNOS-E> linksys mini-certificate 9577 9577-display “23:05:45
   2004-11-25”
```

Actions

```
Mini Certificate:
OTU3Ny1kdWVyb2QAAAAAAAAAAAAAAAAAAAAAAAAAAAA5NTc3AAAAAAAAAAAAAAAAMDMwNT
   Q1MjYxMTA0ybYgcwG8IeaYz225Grs7sDJflnfyJxARPehQ+CO6WisAZ77U2zBi8TCa
   pIwqcDhNXwgYKZxljAET3dFnzAxs2ze1/
   kEHCqvUmDIEjaYL+1WTySaI1TGKy15FbyZb6dQXtbPF+fXiRP//
   caFfKUBTuuwtjExxaAz0H3u8Tc2YT/wH7a0+snpUTFeK/
   Sv9vd7aAUbufSxewlL2GeTdOu0v2i4R25/
   RH6iOHyChGpVt2EJ3BHAlLgXTfJibiwwkrMSe1grSibsCy0D825ezAt66AVKTa/
   hOmSBvdZvdamJIsbP89vnAJPiOfWNet8T40/wOYyylAE5JDJ/2+G/
   MDyc5ImzFTvifKvIQ55T7Jr5E0RUbacDZIlHy5oW+x4sfawCiQZunnb11qlAgYhvOe
   uo4f3JGUKJAld0GRjHfvjRhb3c=

SRTP Private Key:
Oxq38oJqjhe++yBTtTotoMndnZXulkgnnxFQPd0v96oc81IZ5dug9Szob9ZYQXsPkWAxSb
Oxq38oJqjhe++BVpyxz2P2qtZEg==

NNOS-E> linksys check-mini-cert
   OTU3NwAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA5NTc3AAAAAAAAAAAAAAAAMT
   U1ODQ4MTEyNTA0z1TBkpXzjmR6PFX5K4S7G5SxdpozH460T14KpwOxZ8ly4KWpFlcC
   2rTTWEU6WnOufcj5Bfif7cdsAF/
   89kZu83NFceK2ZBRGrJ4cbxREtuPwy1FqkXpBQcztTFXjeyFaq8K7OESebQayFetBE
   ceIupuzxfedlJPRsMRhsHN1uKpomc/
   tdJFHJhxSzn+fX+GTACrXQEHzI+ooDL+iQvzhJ1zk/
   gXTGuk76lkJG2XLvSvdjTp8RjQX/F5h0GnBa02d3bQ51n7IBvJnTeaGKp/U/
   e5pQvW5u6vD/uHkqkTGkZDZzOyIISIdgWVxdjA9cpaSa2D5nPhr8G/
   WhOadLZ08fmB0kPwEFjJ0h0dojjknjNJp/
   qVjR5NEEzuj5kH7Qlvxk25l0MThhydCYpbxShy2GSno7apnyCA02YBQCRlGBOs=
Certificate has expired

NNOS-E> linksys generate-ca-key
Unable to overwrite Linksys CA key

NNOS-E> linksys display-message

   AAAAAMFgaBarzavNNzc3NwAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA3Nzc3AA
   AAAAAAAAAAAAAAMDQ1OTU5MTEwMTA4qtnZJ1ER7t64CqEHeDYytppxkTkX0OWl6hR8
   RKRCLu5yruwXVM4c4QYJ/
   AjcHZ90vcdyxbEjPXDTO9nZLAZkU8k1iBGUaBUkHZbiae8H3+xa8dIq3m0Ydv28dPq
   A3UkGg3oIepsfaxDJXi/
   ci3VSD+J78hYlVEHwOu0JfI+Y0cuyk1wOxSLIJP7smLTb8oDCuN+wGdMUVNuphM+zY
   6SGJwkaae8sR7sikpun/fG3aFx51aRxSt9AHyAplLZbzPnb
Message ID        : HELLO (0x00000000)
SSRC              : 3244320790 (0xc1606816)
Flags             : 0xabcdabcd
Mini-certificate  :
   User name         : 7777
   User ID           : 7777
   Expiration        : 00:59:59 Sat 2008-11-01
```

```
Public key          :
 0xaad9d9275111eedeb80aa107783632b69a71913917d0e5a5ea147c44a4422eee
 72aeec1754ce1ce10609fc08dc1d9f74bdc772c5b1233d70d33bd9d92c066453
Signature           :
 0xc9358811946815241d96e269ef07dfec5af1d22ade6d1876fdbc74fa80dd4906
 837a087a9b1f6b10c95e2fdc8b75520fe27bf216255441f03aed097c8f98d1cbb2
 935c0ec522c824feec98b4dbf280c2b8dfb019d31454dba984cfb363a48627091a
 69ef2c47bb22929ba7fdf1b7685c79d5a4714adf401f202994b65bccf9db
```

load-balancing

Purpose

Manipulates the load balancing database. Do not execute this action unless instructed to do so by Technical Support.

Syntax

```
load-balancing {rewrite-rules | refresh-interface-state}
```

Example

```
NNOS-E> load-balancing rewrite-rules
Success!
```

location

Purpose

Manages the location cache for the current box. Use the location-database action to manage the database that runs on the master. Select one of the following operations:

- **lookup**—returns location information for the specified AOR.
- **flush**—removes all entries from the location cache. Optionally you can specify whether to flush the cache immediately or as the entries time out. By default, the system flushes entries immediately.
- **delete**—deletes the specified entry from the location cache. Enter the address of record for the entry, which can be found in the **show location-cache** command.
- **restate**—changes the state of the location cache entry. Enter the AOR and the new state.

Actions

- **audit**—verifies that location bindings are not corrupted. You can enter a specific AOR to verify or verify the entire location cache.
- **prune**—removes all or the specified corrupted bindings from the location cache.
- **save**—writes the specified AOR or the entire location cache to the location database on the cluster master.
- **reload**—reloads the location database to the location cache on the local box, wiping out the current cache. If you enter an AOR, only that entry is rewritten.
- **cleanup**—immediately removes all location bindings that are not in a registered state. You can automate this action by setting the **cache-cleanup-interval** property of the location service settings object.
- **expire**—forces the location cache entry expiration time to 0, resulting in no throttling for the following applicable REGISTER request. You can specify the address of record that this action applies to, or apply it to all AORs.

You can also schedule this action as part of routine maintenance using the **task** object.

Syntax

```
location lookup aor
location flush [now | gracefully]
location delete aor
location restate aor {unregistered | trying | in-service | redirect |
   registered | out-of-service}
location audit [aor]
location prune [aor]
location save [aor]
location reload [aor]
location cleanup
location expire [aor]
```

Example

```
NNOS-E> location restate sip:6667778888@abc.com out-of-service
Success!

NNOS-E> location lookup sip:6667770001@best.com
sip:6667770001@best.com currently has the following locations:
Contact:
   <sip:6667770001@192.168.215.95:5060;transport=UDP;line=g4ced3b1>;e
   xpires=59

NNOS-E> location audit
Success!>
```

location-database

Purpose

Manages the location database across the cluster. The primary or master appliance contains the main location database. The external database, which is mirrored from the main database, is the database running on the backup system in a cluster configuration. If a failover happens, the external database becomes the master database. Use the location action to manage the cache that runs on an individual box. Select one of the following operations:

- **merge**—merges the specified file into the existing master location cache. If the merge file has a new binding (i.e., one with an index not present in the cached AOR), it is added to the existing cache. If the merged copy has a binding that already exists in the AOR, the values from the merged copy take precedence, overwriting the values in the existing cache. By default, AA-SBC uses the file /cxc/location.xml. Optionally, you can specify a different file path.
- **replace**—writes the specified file to the location database, wiping out the current entries. By default, AA-SBC uses the file /cxc/location.xml. Optionally, you can specify a different file path
- **save**—writes the location database to the supplied file name. If you do not supply a name, it saves to the default location—/cxc/location.xml.
- **delete**—deletes the specified entry from the location database. Enter the address of record for the entry, which can be found in the **show location-cache** command.
- **flush**—removes all entries from the location database.

You can also schedule this action as part of routine maintenance using the **task** object.

Syntax

```
location-database merge [fileName]
location-database replace [fileName]
location-database save [fileName]
location-database delete aor
location-database flush
```

Example

```
NNOS-E> location-database merge /cxc/backup/location.xml
Success!
```

log-target

Purpose

Turns logging on and off for the current CLI session. You configure the CLI as a log target, including the desired class and severity filters, using the services cli object.

For any given CLI session, you can turn this output on or off using this **log-target** action. You may want to do this, for example, so that you can record logging in one CLI session, while working in another. In that way, your work will not be interrupted by log messages.

You must have enabled CLI logging and event logging on a global level for this action to take effect. Set admin to enabled for both the cli and event-log objects.

This action is not available from the AA-SBC Management System.

Syntax

```
log-target {enabled | disabled}
```

Example

```
NNOS-E> log-target enabled
NNOS-E>
```

login

Purpose

Manages user logins. You can either terminate any active login session or unlock users that have been locked out.

- **kill**—to terminate a session, use the **show login-sessions -v** command to list active sessions with identifiers. Specify the session type and the ID to identify the session to end.
- **unlock**—reinstates access to a user who has been locked out. (This may happen, for example, if the user violated the password-policy configuration.) Specify the name of the user to unlock. Use the **show local-users** command to verify that a lockout is the problem for the user.

Syntax

```
login kill {console | ssh | telnet | web | web-service | desktop |
   monitor} id
login unlock name
```

Example

```
NNOS-E> login kill web 2
Success!
```

loopback

Purpose

Establishes an outgoing SIP loopback call. In this call type, the media is looped back to AA-SBC to provide various performance metrics. The loopback process works by an endpoint encapsulating and reflecting RTP packets back to AA-SBC. Data such as media path, round-trip time, and packet loss are written to the AA-SBC database, and can be displayed using the AA-SBC Management System **Call Logs Monitored Calls** link or the **call-loopback** Trend Graph (on the **Status** page). To execute this action, specify which type of loopback to perform and other parameters:

- **packet**—AA-SBC initiates sending RTP packets. When the endpoint receives an RTP packet it responds by reflecting it back to AA-SBC.
- **packet-init**—when AA-SBC initiates the action, the endpoint begins responding immediately by sending packets of the type negotiated in the SDP. When the endpoint receives the RTP packets, it ceases sending SDP packets and reflects RTP back to AA-SBC.
- ***seconds***—specify the duration of the call, between 2 and 600 seconds.
- ***to***—specify the endpoint in the format of a SIP URI.
- ***from***—optionally, specify the endpoint in the format of a SIP URI.
- ***protocol***—optionally, specify the content of the FROM header in the SIP INVITE.

You can also schedule this action as part of routine maintenance using the **task** object or from the **Monitored Calls** or **Monitored URIs** pages in the AA-SBC Management System.

Syntax

```
loopback {packet | packet-init} seconds to [from] [any | udp | tcp |
   tls]
```

Example

```
NNOS-E> loopback packet 10 sip:5554443211@jane.cov.com
Success!
```

make-usb-bootable

Purpose

Changes the state of a USB stick that was used for a system upgrade. This action is only necessary on those platforms that require removal of the USB stick in order for the AA-SBC device to boot from its hard drive. In those cases, you must remove the stick prior to the system rebooting. By doing so, you leave the stick in an unusable state. However, the stick still contains data required for the upgrade (license, configuration, etc.). To retrieve that information:

1. Start the upgrade and remove the stick from the AA-SBC device before the system reboots.
2. After the system reboots, reinsert the stick.
3. Execute this action with the **true** option.
4. Perform a warm restart to retrieve additional configuration from the stick.

Do not use the **false** option with this action unless instructed to do so by Technical Support.

Syntax

```
make-usb-bootable [true | false]
```

Example

```
NNOS-E> make-usb-bootable true
Success!
```

media-delete

Purpose

Deletes all media files associated with a specific session ID. Use the **show media-files** command to list the files. The path field of the output is the complete path and file name. That name also contains the session ID.

Syntax

```
media-delete sessionID
```

Example

```
NNOS-E> show media-files
session-id           channel  date                         path
----------           -------  ----                         ----
0x4c2264e08bc7098    0        15:30:08 Thu 2006-02-23    /cxc_common/
   rtp_recorded/04c2/264e/08bc/sess-04c2264e08bc7098-0-0.xml
0x4c2264e08bc7098    1        15:30:08 Thu 2006-02-23    /cxc_common/
   rtp_recorded/04c2/264e/08bc/sess-04c2264e08bc7098-0-1.xml
0x4c2264e09522afa    0        15:30:17 Thu 2006-02-23    /cxc_common/
   rtp_recorded/04c2/264e/0952/sess-04c2264e09522afa-0-0.xml
0x4c2264e09522afa    1        15:30:17 Thu 2006-02-23    /cxc_common/
   rtp_recorded/04c2/264e/0952/sess-04c2264e09522afa-0-1.xml

NNOS-E> media-delete 0x4c2264e08bc7098
Success!
```

media-delete-old

Purpose

Invokes AA-SBC to delete all recorded media files that are older than the specified number of days or seconds. Enter a number and a unit of measure. By default, AA-SBC deletes media files older than seven days when you execute this command.

You can also schedule this action as part of routine maintenance using the **task** object.

Syntax

```
media-delete-old age [days | seconds]
```

Example

```
NNOS-E> media-delete-old 30 days
Success!
```

media-directory-clean

Purpose

Removes empty recorded media directories. You may have an empty directory, for example, if AA-SBC cleaned a directory as part of a scheduled maintenance operation. That action removes data but leaves the directories. You can also schedule this action as part of routine maintenance using the **task** object.

Syntax

```
media-directory-clean
```

Example

```
NNOS-E> media-directory-clean
Success!
NNOS-E>
```

media-package

Purpose

Creates a tarball of specified WAV files suitable for installation on other AA-SBC devices. Use a wildcard expression to enter multiple source files. AA-SBC verifies that all files are playable WAV files. Enter the desired tar ball file name as the destination.

Syntax

```
media-package sourceFiles destinationFile
```

Example

```
NNOS-E> media-package /cxc_common/media/*.wav package.tar.gz
Success!
```

media-session-audit

Purpose

Sets the minimum age for a media session before it is subject to a media session audit. When you execute this action, AA-SBC checks to ensure that each media session has a matching signaling session. If AA-SBC finds a media session without accompanying signaling, it deletes the orphaned media session. Because the media session is established before signaling is set up, AA-SBC disregards sessions until they have reached an age set by this action. See Setting time and time intervals for information on entry format requirements for minimum age.

Syntax

```
media-session-audit minAge
```

Example

```
NNOS-E> media-session-audit 2:30
Success!
```

mikey

Purpose

Provides utilities to work with the Multimedia Internet KEYing (MIKEY) key management scheme. MIKEY is intended for use with real-time applications, specifically to set up encryption keys for multimedia sessions that are secured using SRTP. (See *RFC 3830, MIKEY: Multimedia Internet KEYing*, for more information.) Select one of the following options:

- **display-message**—displays the base-64 encoded MIKEY messages sent in SDP. For encrypted fields (mainly the KEMAC field), the system displays hexadecimal values within the MIKEY message. Both the field values and the derived SRTP Master keys are displayed for each direction (if the KEMAC is not encrypted). Enter the content of the a=key-mgmt:mikey field. The **derived-salt** Boolean indicates whether the SRTP Master Salt (typically the last 14-bytes of the key) is directly used (**false**) or derived from the MIKEY message (**true**). The default is **false**.

- **decrypt-message**—decrypts the encrypted portions of the MIKEY message and then displays the base-64 encoded MIKEY messages sent in SDP. If the encryption algorithm in the KEMAC field is not enc-alg=None, use this option to decrypt the field. Once decrypted, the action then provides the same display as the output of the **display-message** option. For this action to work, the data-type field of the MIKEY message must be Pre-Shared Secret. Enter the content of the a=key-mgmt:mikey field and the pre-shared key (secret) associated with the message. The **derived-salt** Boolean indicates whether the SRTP Master Salt (typically the last 14-bytes of the key) is directly used (**false**) or derived from the MIKEY message (**true**). The default is **false**.

Syntax

```
mikey display-message message [true | false]
mikey decrypt-message message secret [true | false]
```

Example

```
NNOS-E> mikey display-message
   AQAVgCSvwBQCAAAAAAAAAAAAAADmKwoBAAAAAAUBAAVtaWtleQsAw6WosRcKPAAKFB
   DQ7YekSuOfDzXIDepCWouYsdFsAQAAADYCAQEDBAAAAKAEBAAAAHALBAAAAFAAAQEB
   BAAAAIAJAQAGAQAFAQAIAQEKAQEHAQEMBAAAAAAAAAAkABAAECLFFMcmhxxR1ssTIa
   W5Zw8ADrQBG2lw7G64M2W3PTLwAA==
MIKEY Message:
Version           : 1 (0x01)
  Data-type         : Preshared - Init (0x00)
  Response          : true
  Pseudo-Random     : MIKEY-1 (0x00)
  CSB-ID            : 615497748 (0x24afc014)
  CS Maps           : 2 SRTP-ID maps
     Policy            : #0, SSRC=0 (0x00000000), ROC=0
     Policy            : #0, SSRC=3861580289 (0xe62b0a01), ROC=0
  Attributes        : 5 attributes
     Extension (0x15)          : SDP (01), len=5, 0x6d696b6579
     Timestamp (0x05)          : NTP-UTC (00), 0xc3a5a8b1170a3c00
     Random (0x0b)             : 20-byte
   random=0x10d0ed87a44ae39f0f35c80dea425a8b98b1d16c
     Security Policy (0x0a)    : #0: SRTP, total-length=54 bytes,
   Auth-alg (2)=0x01, Auth-key-len (3)=0x000000a0, Salt-key-len
   (4)=0x00000070, Auth-tag-len (11)=0x00000050, Enc-alg (0)=0x01,
   Enc-key-len (1)=0x00000080, FEC (9)=0x00, KDR (6)=0x00, PRF
   (5)=0x00, Rtcp-enc (8)=0x01, Rtp-auth (10)=0x01, Rtp-enc (7)=0x01,
   Prefix-len (12)=0x00000000
     Keymac (0x01)             : enc-alg=None, 36-byte
   value={key-type=TGK-Salt, key-validity=None, length=16,
   value=0x22c514c726871c51d6cb1321a5b9670f, 14-byte salt
   value=0xb4011b6970ec6eb83365b73d32f0}, mac-alg=None
```

mix-session

Purpose

Mixes audio files into a WAV file. You can use the file-play action once you have created the file.

When AA-SBC records a call, each direction is stored in an XML format. The XML format contains information about packet timing etc. (SSRCs, timestamps, sequence numbers). The **mix-session** action takes the audio data out of those XML files (usually two files—one for each direction) and puts it into a single .WAV file. During that process, AA-SBC decodes (if necessary) to a standard linear format, which you select.

Enter:

- ***sessionID***—the session ID of the originating audio file. Use the **show media-files** command to list the files and find the ID.
- ***fileName***—the WAV destination file. AA-SBC creates a file with just the name you supply, so append the **.wav** suffix to the file name if you want it.
- ***outputChannels***—the number of channels the file should be mixed for. Enter 1 for mono, 2 for stereo. If you specify stereo, you will hear the different sides of the conversation through different speakers. The default setting is 2.
- ***WAV format***—the format you'd like the final WAV file in. The default format is **pcmu**.
- ***recordedPath***—Specifies the location of the files to be mixed. Use this only if the files are not in the default location.

Syntax

```
mix-session sessionID fileName [outputChannels] [pcmu | pcma | pcm16]
    [recordedPath]
```

Example

```
NNOS-E> mix-session 0x4c22760ab06a58a test1.wav
Success!
```

mos-calculate

Purpose

Calculates a MOS score based on various network conditions. Mean Opinion Score (MOS) is a subjective measurement and an “opinion” of the audio quality heard by the listener on a phone. By plugging values into this action that represent your network conditions, you can determine the call quality. See the *Net-Net OS-E – Session Services Configuration Guide* for more information on formulating MOS results.

Enter the following parameters:

- ***pktsReceived***—the number of RTP packets received.
- ***pktsLost***—the number of packets lost during a call, detected by RTP header examination.
- ***pktsDuplicated***—the number of duplicate RTP packets during a call, detected by RTP header examination.
- ***jitter***—the average jitter, in milliseconds.
- ***latency***—the average transmission delay for each packet, in milliseconds.
- ***codec***—the CODEC used which to base default values for other fields.
- ***Rfactor***—the R-factor, a rating on the overall conversational quality of a call, expressed on a 0-to-100 scale. A value of 0 uses the CODEC defaults, otherwise enter a number up to 100, where 0 is extremely bad quality, and 100 is very high quality.
- ***pktInterval***—the ptime (nominal time between adjacent RTP packets) for the CODEC, in milliseconds. Enter a value between 0 and 100. A value of 0 indicates the default ptime for the CODEC.

Syntax

```
mos-calculate pktsReceived pktsLost pktsDuplicated jitter latency
  codec [Rfactor] [pktInterval]]
```

Example

```
NNOS-E> mos-calculate 10 0 3 6

Codec                    : pcmu
Packet-interval          : 20 msecs
R-factor                 : 93.2
Received                 : 7 packets
Lost                     : 0 packets

Average Jitter           : 6 msecs
Average Latency          : 0 msecs

MOS                      : 4.44


NNOS-E> mos-calculate 10 1
Codec                    : pcmu
Packet-interval          : 20 msecs
R-factor                 : 93.2
Received                 : 10 packets

Lost                     : 1 packets
Average Jitter           : 0 msecs
Average Latency          : 0 msecs
MOS                      : 4.06
```

mount

Purpose

Mounts a data partition on the selected drive so that AA-SBC can access the data on the device. You can also mount a CD-ROM or USB device. Use this action to insert a device into a live system. (If the device is present at boot, AA-SBC automatically mounts it.) Use the **show mounts** command to display mount status; the Drive Name field indicates whether a partition is mounted.

Syntax

```
mount {data-1 | data-2 | usb | cdrom | system-1 | system-2}
```

Example

```
NNOS-E> mount hard-drive-1
```

Actions

```
Device is mounted
NNOS-E>
```

orderly-restart

Purpose

Causes a restart of the type specified after gracefully terminating any existing connections. By default, the **orderly-restart** action causes the box to restart at the first point in time when there are no active calls. This is useful for code upgrades on a unit which is currently in service. To immediately restart the box, use the restart action.

Select one of the following restart types:

- **warm**—exits and restarts the AA-SBC application when there are no active calls. This is the default.
- **cold**—exits the operating system and then restarts.
- **halt**—stops AA-SBC and does not restart the system.
- **cnx0**, **cnx1**—stops and restarts the specified CNX card.
- **cluster**—performs a warm restart of all boxes within the cluster.
- **controlled**—performs a warm restart of all boxes within the cluster without interrupting call flow.
- **cancel**—cancels the operations initiated by the **orderly-restart controlled** action.

You can also schedule this action as part of routine maintenance using the **task** object. Use the **show orderly-restart** command to report on the current status of an invoked orderly-restart action (i.e., to display a report of the number of currently active calls that this system is awaiting termination on.)

Syntax

```
orderly-restart {warm | cold | halt | cnx0 | cnx1 | cluster |
   controlled | cancel}
```

Example

```
NNOS-E> orderly-restart cnx0
Are you sure (y or n)? y
Success!
```

```
NNOS-E>
```

performance-tracking

Purpose

For Technical Support use only.

Syntax

```
performance-tracking start [filename] [samplingInterval]
   [collectionDuration] [enabled | disabled]
performance-tracking info
performance-tracking stop
```

ping

Purpose

Tests whether a specific IP address can accept requests, verifying the existence and connectivity of a host on the Internet. Enter a host name or IP address. Optionally, you can set the number of attempts (packets sent) and the outgoing interface used. The default number of packets sent is 3. AA-SBC does a route lookup to select an outgoing interface unless you specify otherwise. You can also verify connectivity using the arp **request** action.

Syntax

```
ping host [count] [interface]
```

Example

```
NNOS-E> ping 196.84.32.1
no response from 196.84.32.1
no response from 196.84.32.1
no response from 196.84.32.1
3 packets sent, 0 packets received, 3 packets lost (100%)

NNOS-E> ping 172.26.0.49
28 bytes from 172.26.0.49: 0.272 ms
28 bytes from 172.26.0.49: 0.236 ms
28 bytes from 172.26.0.49: 0.201 ms
```

Actions

```
3 packets sent, 3 packets received, 0 packets lost (0%)
roundtrip minimum/average/maximum: 0.201/0.236/0.272 ms
```

playback

Purpose

Places a call to the specified SIP URI, plays the recorded media specified by the session ID, and then disconnects the call. Compare this to the file-play action. The **playback** action plays recorded sessions only (AA-SBC takes care of mixing the media for playing). The file-play action plays any file. For example, if you made a file using the mix-session action, you can play it using file-play.

Enter the following information:

- ***sessionID***—the session ID of the recorded media. Use the **show media-files** command to list the files and find the ID.
- ***to***—the SIP URI that specifies where to place the call to.
- ***from***—optional. A SIP URI that appears as the caller ID.
- ***transport***—optional. The transport protocol to use, either any, UDP, TCP, or TLS.

Syntax

```
playback sessionID to [from] [transport]
```

Example

```
NNOS-E> playback 0x4c22760ab06a58a sip:management@cov.com
   sip:hr@cov.com
Success!
```

presence

Purpose

Manages the presence cache. The primary or master appliance contains the main presence cache. The external cache, which is mirrored from the main cache, is the database running on the backup system in a cluster configuration. If a failover happens, the external cache becomes the master cache. See the external-presence action for information on managing the external cache.

Select one of the following operations:

- **merge**—merges the specified file into the existing master presence cache. If the merge file has a new URL entry, it is added to the existing cache. If the merged copy has a URL that already exists, the values from the merged copy take precedence, overwriting the values in the existing cache.
- **replace**—writes the specified file to the presence cache, wiping out the current cache.
- **save**—Writes the presence cache to the supplied file name. If you do not supply a name, it saves to the default location—/cxc/presence.xml.
- **delete**—deletes the specified entry from the presence cache. Enter the URL for the entry, which can be found in the **show presence-cache** command.
- **flush**—removes all entries from the presence cache.

You can also schedule this action as part of routine maintenance using the **task** object.

Syntax

```
presence {merge fileName | replace fileName | save fileName | delete
   URL | flush}
```

Example

```
NNOS-E> presence delete 5085551212@abc.com
Success!
```

prune-assoc

Purpose

Immediately removes inactive associations from the location database to reclaim memory. You can also configure this to happen at a regular interval by enabling the prune-association property and defining the frequency with the **pruning-interval** property, both in the settings object.

Syntax

```
prune-assoc
```

Example

```
NNOS-E> prune-assoc
Success!
```

radius

Purpose

Enables, disables, or tests a previously configured server that is part of a RADIUS group. Enter a reference to the configured server (configured with the RADIUS group server object). Enclose the reference path in quotation marks, and keep in mind that the server name is case-sensitive.

When using the **test** action, you can validate user credentials on the server via AA-SBC. When invoked, AA-SBC sends a test authentication message to the server to ensure that the RADIUS server is configured properly. The RADIUS server has a list of users and their associated passwords. This action verifies the name and, optionally, password, as well as the Digest settings for the user. Enter the following:

- ***radiusServerReference***—a reference to the configured server and group. Enter this in quotation marks in the format “*groupNamePath*\server *ipAddress*”
- ***userName***—the user name to test. This is the name configured on the RADIUS server.
- ***password***—optional. The password associated with the specified user name, as configured on the RADIUS server.

- ***digest***—optional. The setting for Digest use for the specified user name. Enter **true** (user sends Digest requests) or **false**. The default is true.
- ***digestRealm***—optional. The realm the Digest user is associated with. The default setting is **testrealm**, which is recognized and accepted by most RADIUS servers.

Syntax

```
radius deactivate radiusServerReference
radius reactivate radiusServerReference
radius test radiusServerReference userName [password] [true | false]
   [digestRealm]
```

Example

```
NNOS-E> radius reactivate “vsp radius-group East server Boston”
Invalid object
NNOS-E> radius reactivate “vsp radius-group East server boston”
Success!
NNOS-E>
```

The following examples illustrate the **radius test** action. In the first example, the test succeeds because the Digest

```
NNOS-E> radius test “vsp\radius-group East\server 172.26.0.147" user1
   password1 false
RADIUS test authentication:
 User name: user1
 Password: password1
 Type: Normal
 Result: Accept
```

The following RADIUS test fails because it is a Digest request (Type=Digest) and the user is not configured on the RADIUS server for Digest:

```
NNOS-E> radius test “vsp\radius-group East\server 172.26.0.147" user1
   password1

RADIUS test authentication:
 User name: user1
 Password: password1
 Type: Digest
 Realm: testrealm
 Result: Reject
Authentication attempt failed
```

raid-check-consistency

Purpose

This action only applies to the NN 2620 with RAID controller. Starts or stops a consistency check on the specified RAID logical volume. The consistency check compares images on mirrored or mirrored/striped drives, and reports inconsistencies to the event log. (The RAID controller automatically manages resynchronization of inconsistent logical volumes.)

Syntax

```
raid-check-consistency {start | abort} {L0 | L1}
```

Example

```
NNOS-E> raid-check-consistency start L0
Invalid provider
```

raid-set-adapter

Purpose

These actions only apply to the NN 2620 with RAID controller. Sets and manages thresholds, rebuild rates, and refresh intervals for the RAID controller. In addition, you can control (silence) the audible alarm.

Select one of the following operations:

- **alarm**—enables or disables the audible alarm that indicates that a logical volume is not optimal (e.g., physical drive dead, logical drive out of sync). Set to **enabled** to activate the alarm; **disabled** deactivates it. The **silence** option stops the sound of the current alarm, leaving the feature enabled.
- **cache-flush-interval**—sets the number of seconds between flushes of the RAID controllers battery cache. The option sends the contents of the battery cache memory to the logical drives. Enter a value between 0 and 255.
- **rebuild-rate**—sets the percentage of the compute cycles that are dedicated to rebuilding data onto a new physical disk after a drive has failed.

- **patrol-read-rate**—sets the percentage of the compute cycles that are dedicated to preventative scanning. A patrol read scans the system for possible physical disk drive errors that could lead to drive failure. It helps protect data integrity by taking corrective action on the error before failure occurs.
- **cc-rate**—sets the percentage of the compute cycles that are dedicated to a consistency check of data across logical volumes.
- **recon-rate**—sets the percentage of the compute cycles that are dedicated to reconstruction (resynchronization and copy) operations. This activity is undertaken automatically by the RAID controller if the logical volumes are not synchronized.
- **pred-fail-poll-interval**—sets the number of seconds between polls of the hard drives for reliability status. Enter a value between 0 and 65535.
- **battery-warn**—enables or disables the battery warning message displayed in BIOS. This is the battery backup capability for the RAID array. If **enabled**, when BIOS boots, if it detects low or no battery, it displays a message to the screen.

Syntax

```
raid-set-adapter alarm {enabled | disabled | silence}
raid-set-adapter cache-flush-interval seconds
raid-set-adapter rebuild-rate percentage
raid-set-adapter patrol-read-rate percentage
raid-set-adapter cc-rate percentage
raid-set-adapter recon-rate percentage
raid-set-adapter pred-fail-poll-interval seconds
raid-set-adapter battery-warn {enabled | disabled}
```

Example

```
NNOS-E> raid-set-adapter alarm silence
Success!
```

reg-lookup

Purpose

Displays, for a specified URI, the registration-plan settings that AA-SBC assigned, the routing arbitration process, and the selected server.

This action simulates the registration routing path, but does not actually trigger an outbound call. It exercises both registration plan and location database lookups. Its output indicates which registration plan entry (or location cache entry) the registration would use and the next hop.

Enter a To URI. The action returns results for any AOR that matches a configured registration plan, or you can find the URI of interest in the AOR field displayed with the **show location-cache** command.

Syntax

```
reg-lookup toUri [soureceIP] [localPort]
```

Example

```
NNOS-E> reg-lookup sip:4135555555@company.com
Resulting priority 200 sequential hunting total 2 next 0

All matching routes:
route phone 413555!* priority 200 best yes
This call will be forwarded to 12.39.208.251 transport UDP port 5060
```

reg-lookup-detail

Purpose

Displays, for a specified To URI, the content of the session configuration associated with the selected registration-plan. If the SIP URI matches a registration-plan (source-) route or arbiter, the action returns the session configuration for that entry (or the default session configuration if the entry does not have one assigned). If the registration-plan **route** has the **peer** set to server, and the server has a session configuration, this action displays the merged session configuration. (The server session configuration takes precedence over the **route** session configuration.)

Syntax

```
reg-lookup-detail toUri [soureceIP] [localPort]
```

Example

```
NNOS-E> reg-lookup-detail sip:2078548355@elmaple.com
Resulting session config merged from server/group egress:
sip-settings
mode auto-determine
transport any
port auto-determine
route-hdr none
route-hdr-use-fqdn enabled
route-hdr-uri-host
route-hdr-add-register-msg disabled
route-hdr-preprocess-strip disabled
lcs-compatibility disabled
in-server unknown
out-server unknown
utilize-contact enabled
add-contact-nat disabled
compress-signaling disabled
preserve-call-id disabled
preserve-cseq disabled
proxy-generate-100-trying
handle-3xx-locally enabled
handle-3xx-locally-lookup-original-invite disabled
session-timeout 300 seconds
session-duration-max 0 seconds
--more--
```

registration

Purpose

Manages the registration information for AA-SBC, specifically the registration routing and client tables. Use the **show registration-clients** and **show registration-routing** commands to view bindings and statistics for the tables.

Enter one of the following:

- **refresh**—resends a REGISTER for each entry to reset the client binding. AA-SBC sends the register to a specific peer, if specified, or all peers if not. To send only to a specific peer, enter the SIP URI set in the server **peer-identity** property.
- **purge**—clears all entries from the registration client table.
- **query**—sends a REGISTER to the specified peer with a query regarding a specified peer. The peer returns its current binding. Enter the address of record you are interested in and the server to which AA-SBC should send the request.
- **proxy**—simulates the registration proxy process. Instead of sending a registration on behalf of AORs in the location database, the registration proxy sends a registration manually for the specified AOR.
- **client-save**—restarts collection of client state information if collection was stopped using the **client-clear** option.
- **client-clear**—purges the state information from the client table and stops the future collection of client state data. Use **client-save** to restart client state collection.
- **status-clear**—clears all registration status counters. These are the entries that can be viewed with the **show registration-status**, **show registration-delegate-status**, and **show registration-proxy-status** commands.
- **provider-enabled**—enables the location information services provider.
- **provider-disabled**—disables the location information services provider.

Syntax

```
registration refresh [peerURI] |
registration purge [peerURI]
registration query AORpeer peerURI
registration proxy AORpeer peerURI fromURI contactHeader
registration client-save
registration client-clear
registration status-clear
registration provider-enabled
registration provider-disabled
```

Example

```
NNOS-E> registration client-save
Success!
NNOS-E> registration refresh
Success!
```

remove-device

Purpose

Removes the specified device from the list of devices that automatically mount at boot-up. Use this action before physically removing a drive from the system. Before executing this action, ensure that you do not have any configuration pointing to mounts that are on the removed device (e.g., recorded files, logs), as writes to that device will fail.

Syntax

```
remove-device {data-1 | data-2}
```

Example

```
NNOS-E> remove-device data-2
Are you sure (y or n)? y
Changes will take effect at the next restart
NNOS-E>
```

restart

Purpose

Causes an immediate restart of the type specified. To restart after gracefully terminating any existing connections, use the orderly-restart action. Select one of the following restart types:

- **warm**—exits and restarts the application.
- **cold**—exits the operating system and then restarts.
- **halt**—stops AA-SBC and does not restart the system.
- **cnx0**, **cnx1**—stops and restarts the specified CNX card.
- **cluster**—performs a warm restart of all boxes within the cluster.
- **controlled**—restarts members in a cluster without interrupting call flow.
- **cancel**—Cancels the operations initiated by the **restart controlled** action.

The default type is **warm**.

You can also schedule this action as part of routine maintenance using the **task** object.

Syntax

```
restart {warm | cold | halt | cnx0 | cnx1 | cluster | controlled |
   cancel}
```

Example

```
NNOS-E> restart warm
Are you sure (y or n)? y
NNOS-E is restarting...

NNOS-E>

NNOS-E> restart controlled
Are you sure (y or n)? y
restarting 172.66.0.10...
restarting 172.66.0.11...
restarting 172.66.0.12...
NNOS-E is restarting...
NNOS-E>
```

restore-defaults

Purpose

Resets AA-SBC configuration settings to the factory defaults and executes a cold restart of the system. Use this with care, as your startup configuration is deleted.

Syntax

```
restore-defaults
```

Example

```
NNOS-E> restore-defaults
Are you sure (y or n)? y
NNOS-E is restarting...
```

restore-stick-create

Purpose

Creates a bootable USB recovery stick by copying system images to a USB stick plugged into the USB port of the system. When you select the **full-backup** option, the default, the current image on AA-SBC, including application and configuration files, and all associated software, is written to the stick. The recovery image does not include copies of the AA-SBC database, system tar files (.gz), or of media files on the system at the time of creation. When you select the **config-backup** option, just the current configuration file is written to the stick.

Determine how often to create/update the recovery stick based on the frequency of configuration changes to your system. capture the current software, certificates, and operating system image to the USB stick.

Syntax

```
restore-stick-create {full-backup | config-backup}
```

Example

```
NNOS-E> restore-stick-create full-backup
Starting rescue-stick-create as a background operation.
 -- this could 10 minutes or longer --
```

Actions

```
Please use the USB stick's activity light as an indication when this
   operation is complete.
```

rtp-cache-delete

Purpose

Deletes the specified WAV file or DTMF event from the RTP cache, if the entry is inactive. AA-SBC caches copies of encoded media stream data for music-on-hold and periodic/introduction announcements, instead of re-encoding them for each call. However, if the configuration no longer uses an announcement or if use of a particular CODEC is discontinued, you can free up memory resources occupied by unused announcements by cleaning the cache of these unused entries. AA-SBC checks to make sure the entry is not currently streaming to an on-hold call.

Use the **show rtp-cache** command output to find the required file name or event ID. Delete based on either:

- **file**—enter the file name of the WAV file to delete.
- **event**—enter the event ID for the entry, and optionally, the volume. DTMF events display as “Event=event_id/volume” in the **show rtp-cache** output.

Syntax

```
rtp-cache-delete file filename [codec] [packetTime]
rtp-cache-delete event eventID [volume] [codec] [packetTime]
```

Example

```
NNOS-E> rtp-cache-delete file announce1.wav
Success!

NNOS-E> show rtp-cache
Type  Cache  Codec    PacketTime Current-Streams Pkts-Cached Pkts-Sent
----  -----  -----    ---------- --------------- ----------- ---------
event  0/10   g723     30         0               33          0
              g728     20         0               50          0
              g729     20         0               50          0
              iLBC     30         0               33          0
              pcma     20         0               50          0
                       30         0               33          0
              pcmu     20         0               50          0
                       30         0               33          0
              g7221    20         0               50          0
              g726-16  20         0               50          0
              g726-24  20         0               50          0
              g726-32  20         0               50          0
              g726-40  20         0               50          0
              gsm-amr  20         0               50          0
       1/10   gsm      20         0               50          0
              g723     30         0               33          0
              g728     20         0               50          0
              g729     20         0               50          0
              iLBC     30         0               33          0

NNOS-E> rtp-cache-delete event 0 10
Success!


NNOS-E> show rtp-cache
Type   Cache   Codec     PacketTime Current-Streams Pkts-Cached Pkts-Sent
----   -----   -----     ---------- --------------- ----------- ---------
event   1/10    gsm       20         0               50          0
              g723     30         0               33          0
              g728     20         0               50          0
              g729     20         0               50          0
              iLBC     30         0               33          0
```

rtp-header

Purpose

Decodes binary RTP packets (the output of the srtp action). The action results in output that prints out the fields of the header, as defined in RFC 1889, RTP: A Transport Protocol for Real-Time Applications. To execute, enter the hexadecimal value of the binary RTP header. The action returns

- the version and any additional flags
- RFC-defined payload type
- the sequence number, used to detect missing or out-of-order packets
- the timestamp, a sample count used to synchronize RTP sequences
- the source synchronization ID.

Syntax

```
rtp-header packet-header
```

Actions

Example

```
NNOS-E> rtp-header 0x8070c351ed6508afff311f7b
Version       : RFC-1889
Payload type  : 112 (0x70) Sequence     : 50001 (0xc351) Timestamp
   : 3982821551 (0xed6508af) SSRC         : 4281409403 (0xff311f7b)
```

rtp-stream

Purpose

Displays the packets that make up a recording. The options allow you to control the detail level of the display and also to create an XML file from a WAV file.

Select either:

- **details**—displays a full or compressed list of all packets in the RTP stream that made up the recording. Enter a file name that contains the recording. You can also select to display timestamps. Select **delta** to display the time passed since the last packet. Select **relative** to display timestamps relative to the beginning of the call. Select **absolute** to display the complete date/time string. By default (**none**) the output displays no time stamps.

  The first true/false boolean controls display of an interpretive line highlighting potential changes. If **true**, the default, the output includes the line. The second true/false boolean controls compression. If **true** (the default), the output only displays a summary of packets as long as the sequence is as expected (no change to SSRC). Each unexpected line is displayed. If set to **false**, the output lists all packets.
- **summary**—displays high-level (header) information that summarizes the packets in the RTP stream.
- **stats**—displays MOS and other information for recorded files, including the CODEC in use, packet counts, jitter and others.
- **create**—creates and encodes an XML file from a WAV file. Optionally you can set the packetization rate.

Syntax

```
rtp-stream details xmlSource {none | delta | relative | absolute} [true
   | false] [true | false]
rtp-stream summary xmlSource
```

```
rtp-stream stats xmlSource
rtp-stream create wavSource xmlDest codec [packetTime]
```

Example

```
NNOS-E> rtp-stream summary /cxc_common/recorded/
   sess-04c2a2312f70ca7a-0-1.xml
Filename: /cxc_common/recorded/sess-04c2a2312f70ca7a-0-1.xml
Start Time: 15:01:56.458930 Fri 2007-07-13
Packet-time: 20
2 rtpmaps:
  PCMU payload-type=0, sample-rate=8000, channels=1   telephone-event
   payload-type=101, sample-rate=8000, channels=1
Success!

NNOS-E> rtp-stream details /cxc_common/recorded/
   sess-04c2a2312f70ca7a-0-1.xml relative
1     34 RTP Invalid: RTP version is invalid
***** SSRC changed from 0 to 1067843479 ******
     2    134 Payload type=PCMU, SSRC=1067843479, Seq=43432,
   Time=1373220824, Mark, payload bytes=160
     3-162  Payload type=PCMU, SSRC=1067843479
   163   3352 Payload type=PCMU, SSRC=1067843479, Seq=43593,
   Time=1373246584, payload bytes=160
***** Missed 1 sequence numbers ******
   164   6679 Payload type=PCMU, SSRC=1067843479, Seq=43595,
   Time=1373273064, payload bytes=160
   165   6700 Payload type=PCMU, SSRC=1067843479, Seq=43596,
   Time=1373273380, payload bytes=160
***** Timestamp discontinuity (possible silence): old(1373273380) +
   expected(316) != new(1373273540) ******
   166   6721 Payload type=PCMU, SSRC=1067843479, Seq=43597,
   Time=1373273540, payload bytes=160
   167-389  Payload type=PCMU, SSRC=1067843479
   390  11197 Payload type=PCMU, SSRC=1067843479, Seq=43821,
   Time=1373309380, payload bytes=160

NNOS-E> rtp-stream stats /cxc_common/rtp_recorded/04c30cd29/
   sess-04c30cd2909dcc70-0-1.xml
Codec           : PCMU (pt=0)
SSRC            : 2691206733 (0x4d8e68a0)
Duration        : 8.74 seconds
Packet interval : 20 msecs
Jitter (sum)    : 116.0 msecs
Received        : 437
Lost            : 0
MOS             : 4.39
```

rule-failover

Purpose

Deletes an individual entry from or flushes the rule failover database. This database contains rules that are internally created by the third-party call control process to do advanced call control operations. You can list current rules with the **show automatic-rules** command. This action is for Technical Support use only.

Syntax

```
rule-failover {delete entry | flush}
```

Example

```
NNOS-E> rule-failover delete rule-24
Success!
```

script

Purpose

For Technical Support use only.

Syntax

```
script filename [variable1] [variable2] [variable3] [variable4]
   [variable5] [variable6] [variable7] [variable8] [variable9]
```

secret

Purpose

Manages AA-SBC passwords and tags. AA-SBC uses this two-part password mechanism for passwords shared with other devices (also known as shared secrets). See Understanding passwords and tags for a complete explanation of this mechanism.

You can also set password/tag associations from various points within the configuration. This password mechanism applies does not apply to passwords created for users under the access object. Use the **show secrets** command to view configured password tags.

Enter one of the following:

- **set**—creates a password/tag association. If you re-execute this action, and supply a different password, AA-SBC overwrites the password that was associated with the tag with the new password. When you set an association, you supply a tag. The system then prompts you for the secret (password). The tag is what users enter, the secret is the password known to the other device. Tags cannot contain the pound symbol (#). If you do not specify a tag, the system saves the password without an associated tag.

Note: You must manually enter passwords on each AA-SBC device. Because passwords are maintained in a separate store, simply copying the configuration file between devices does not copy the password store.

- **delete**—removes a secret so that the association between tag and secret no longer exists.
- **root**—resets the Linux root password. When prompted, specify and confirm the new root password. The root secret must be at least four characters long.
- **ssh**—sets the SSH account password. When prompted, specify and confirm the new password.
- **synchronize**—copies passwords to other devices in the cluster. Passwords are maintained in a separate store; simply copying the configuration file between devices does not copy the password store. Use this action on the master device to copy your passwords the other devices in the cluster.
- **verify**—confirms a secret that is associated with the specified tag. Enter the tag, and the system prompts you for the secret. If you did not associate a secret with a tag, you get a secret mismatch message

Syntax

```
secret set tag secret
secret delete tag
secret root password
secret ssh password
```

Actions

```
secret synchronize
secret verify tag secret
```

Example

```
NNOS-E> secret set red
password: ***********
 confirm: ***********
Success!

NNOS-E> secret verify red
secret: ***********
Success!

NNOS-E> show secrets

tag
---
red
```

send-notify-event

Purpose

Sends an event within the body of a NOTIFY message to the phone at the specified URL. If you select one of the preconfigured events (reboot, resync, restart, or report), AA-SBC sends the event expected by a Sipura phone. To send an event to any other type of phone, enter the appropriate string. Note that some phones and event types require credentials, so username and password may be required. Also, optionally you can specify which interface the NOTIFY message goes out. In the **tag** field, enter a configured **classification-tag** from the **ip** object.

Use this action, for example, to send an event that will reboot the phone, check for configuration changes, and/or download a configuration.

Syntax

```
send-notify-event {string | reboot | resync | restart | report} URL
   [from] [tag] [any | UDP | TCP | TLS] [username] [password]
```

Example

```
NNOS-E> send-notify-event check-sync
   sip:2125551212@voip.acmepacket.com
--- End of Data ---
```

```
SIP Tx: [09:58:23.193602]   457 bytes to 66.10.143.110:22324 on vx1
   (UDP socket 69 - 82.134.77.12:5060):
--- Start of Data ---
NOTIFY sip:2125551212@66.10.143.110;transport=UDP SIP/2.0
From:
   <sip:2125551212@66.10.143.110:22324>;tag=af4da8c0-13c4-469f6dfd-3f
   1eefb-2ac0a04f;rinstance=b6620d67f3884ca8
To: <sip:2125551212@66.10.143.110:22324>;rinstance=b6620d67f3884c
Call-ID: NNOS-E-1-af4da8c0-13c4-469f6dfd-3f1eefb-5f0726f5
CSeq: 1 NOTIFY
Via: SIP/2.0/UDP
   82.134.77.12:5060;branch=z9hG4bK-678d-469f6dfd-3f1eefb-d68961
Event: check-sync
Subscription State: Active
Content-Length: 0
```

sensor

Purpose

Manipulates elements of the sensor management system. Enter one of the following:

- **delete-events**—clears the sensor event log. Use the show sensor-events command to view the contents of the log prior to deleting. The sensor event log maintains information pertaining to such events as temperature, voltage, and others.
- **identify**— activates a blue light on the front of the system that can be used to identify the system you are working with. By executing the sensor identify action, the light will begin to blink and will continue for the number of seconds specified. Enter a value from 1 to 127. The default is 60 seconds.
- **reset-processors**—clears any error or disabled sensor states from all processors in the system. At boot time, the system performs a diagnostic test of its processors. If a processor fails, it is automatically disabled. (The system generates a log event if this happens; also **show sensors** and **show sensor-events** indicates the problem. use this action to begin recovery, followed by the restart **cold** action, causing the processor to be retested during the cold start.

Syntax

```
sensor delete-events
sensor identify [timeout]
sensor reset-processors
```

Actions

Example

```
NNOS-E> sensor identify
Success!
NNOS-E> sensor delete-events
Success!
NNOS-E>
```

server

Purpose

Clears counters and configurations related to the configured servers. Enter one of the following:

- **routing-clear**—clears dial plan and registration plan statistics counters. These are the counters that are visible using the **show dial-plan** and **show dial-plan** commands.
- **purge-dynamic**—for use with configured dns-group servers, purges the server-pool configuration. A dns-group server learns its server-pool configuration dynamically through DNS. If the data is not synchronized, for example the DNS server may have gone down and then come back up, you can execute this action to delete the server-pool configuration and rebuild it through incoming SIP traffic.
- **age-dynamic**—for use with configured dns-group **server**s. When initiated, AA-SBC purges the server IP address, port, and transport protocol information (but maintains the rest of the server configuration). This forces a DNS query the next time the server is used.

Syntax

```
server {routing-clear | purge-dynamic | age-dynamic}
```

Example

```
NNOS-E> server routing-clear
Success!
NNOS-E> server purge-dynamic
Success!
```

service-route-lookup

Purpose

Performs a lookup in the specified service route table. The output of the action returns the best route to the destination, given the filtering parameters applied. It also displays other aspects of the route such as the gateway, physical interface, geolocation, and metrics. The metrics are the resulting values of the services-routing **metric** assignments. For example, if you had assigned user-metric, AA-SBC displays the cost of the route (configured with the ip metric property). If you had assigned intf-throughput, AA-SBC displays the most recent calculation of interface throughput for the route.

You must enter the following arguments:

- **service table name**—selects the specific routing table to search for the best route to the destination. Enter either sip, media, or stun.
- **destination**—specifies the host address of the destination. Enter in IP address format.
- **transport protocol**—for a STUN service routing table lookup, specifies the transport protocol that the route must use to reach the specified destination. Each stun-server in the cluster is configured to support a particular protocol (UDP, TCP, or TLS) with the **port** property. AA-SBC then returns the best route using that protocol.

Optionally you can enter:

- **partner IP address**— specifies the address of a peer in a cluster network. If you use this argument, the output will return the best route from the specified peer. By default, AA-SBC finds the best route from any box (255.255.255.255). Use 0.0.0.0 as the partnerIP address to return results from the local box.
- **load-balance**—selects whether to load balance the results (choose true or false). If set to **true**, AA-SBC uses a round robin algorithm to return results. Each time you execute the command, the next entry is returned. If set to **false**, AA-SBC returns the first entry in the table (the default behavior).
- **geolocation**—filters results based on the geolocation. This value is assigned to an interface with the ip object, and is stored with the route. When you specify a geolocation, AA-SBC returns the best route to the destination that has the specified geolocation.

Actions

Syntax

```
service-route-lookup media destination [partnerIP] [true | false]
   [geolocation]
service-route-lookup sip destination [true | false] [geolocation]
service-route-lookup stun destination {udp | tcp | tls} [true | false]
```

Example

```
NNOS-E> service-route-lookup media 192.168.55.55

service-name:       media
destination:        192.168.55.55/32
gateway:            192.168.215.1
source-ip:          192.168.215.100
interface:          eth0
origin:             local
geo-location:       7654
metric1:            1
metric2:            0
metric3:            0
metric4:            0
metric5:            0
partner-ip-address: 0.0.0.0

NNOS-E>
```

set-call-forwarding

Purpose

Sets AA-SBC to forward any calls intended for the specified address-of-record to a specified URI. Once configured, you can then use the enable and disable arguments to activate the forwarding.

Syntax

```
set-call-forwarding aor {enabled | disabled} callForwardURI [cookie]
```

Example

```
NNOS-E> set-call-forwarding sip:jdoe@cov.com enabled
   sip:confRm1@cov.com
Success!
```

set-chassis-config-boot

Purpose

Sets the system partition from which AA-SBC boots. Use this, for example, to revert to an old system image after having installed and run off of a new one. In that case, set this to the alternate partition and reboot the system. You can use the **on-board-rescue** option to reboot using a limited functionality that provides access to USB stick rescue utilities without using the stick itself. Use the **show chassis-config** command to display the current partition assignment.

Syntax

```
set-chassis-config-boot {system-1 | system-2 | on-board-rescue}
```

Example

```
NNOS-E> set-chassis-config-boot system-2
Changes will take effect at the next cold restart
NNOS-E>
```

set-chassis-config-console

Purpose

Configures the endpoint to which AA-SBC directs console output the next time that it boots. To display output to the screen, set this to the port used for your management console. Use the **show chassis-config** command to display the current management console assignment.

Syntax

```
set-chassis-config-console {serial-0 | serial-1 | vga}
```

Example

```
NNOS-E> set-chassis-config-console system-2
Changes will take effect at the next cold restart
NNOS-E>
```

set-chassis-config-ipmi

Purpose

Enables and disables IPMI functionality. By default, IPMI functionality is enabled, and allows system reports such as those returned by the **show sensors** command. Disable this only in cases where the IPMI functionality causes a problem with your specific platform. Use the **show chassis-config** command to view the current IPMI setting.

Syntax

```
set-chassis-config-ipmi {enabled | disabled}
```

Example

```
NNOS-E> set-chassis-config-ipmi disabled
Changes will take effect at the next cold restart
NNOS-E>
```

set-do-not-disturb

Purpose

Sets AA-SBC to return a busy response to any call directed to the specified address of record. The phone registered to that AOR will respond according to its configuration (busy, voice mail, etc.). Once configured, you can then use the enable and disable arguments to activate and deactivate the setting.

Syntax

```
set-do-not-disturb aor {enabled | disabled}
```

Example

```
NNOS-E> set-do-not-disturb sip:jdoe@cov.com
Success!
```

sip

Purpose

Performs actions on SIP transport connections. Each action, and its specific syntax, is described below.

- **ping**—Pings the specified server, using the SIP OPTION message (instead of ICMP), to validate a SIP node. Enter a transport protocol and a port, if desired. The default protocol is UDP; the default port is 5060.
- **ping-aor**—Pings the specified address of record, using the SIP OPTION message (instead of ICMP), to validate a SIP node.
- **traceroute**—Sends SIP OPTION message trace packets to determine a routing path. The default protocol is UDP; the default port is 5060.
- **server-monitor**—Adds the specified server to the monitor pool. AA-SBC periodically pings the servers in this pool to check availability. Use the **show sip-server-availability** command to view the results for all monitor pool entries. Enter a server hostname or IP address, and optionally, a transport protocol and port. The default protocol is UDP; the default port is 5060.
- **server-unload**—Removes the specified server from the monitor pool (see **server-monitor**, above). Enter a server hostname or IP address, and optionally, a transport protocol and port. The default protocol is UDP; the default port is 5060.
- **lookup-connection**—Does a lookup in the SIP connection table. Specify the endpoint IP address. AA-SBC returns an entry if there is a connection between that endpoint and AA-SBC. Optionally, you can set the transport protocol and/or a local IP address to perform the lookup from, rather than from AA-SBC.
- **delete-connection**—Deletes the connection between AA-SBC and the specified endpoint. Optionally, you can set the transport protocol and/or a local IP address to perform the lookup from, rather than from AA-SBC.
- **purge-connection**—Disconnects and deletes all entries in the connection table. (This is all active calls on AA-SBC.)
- **reset-connection**—Tears down and then resets all entries in the connection table.
- **clear-statistics**—Clears all counters associated with the connection table.

- **ping-resume**—Resumes sending SIP OPTION message ping packets to entries in the monitor pool. These packets would have been stopped with the **ping-suspend** option, below.
- **ping-suspend**—Temporarily halts sending SIP OPTION message ping packets to entries in the monitor pool. AA-SBC uses the last known state for each server as its current state data. Use the **ping-resume** option, above, to restart sending packets.
- **udp-log-on**—Starts logging a partial header from each SIP UDP message to the UDP buffer. The UDP buffer is a FIFO buffer, the size of which is set with the vsp object **max-udp-outbound-log** property.
- **udp-log-off**—Turns off logging of SIP UDP messages to the UDP buffer. See the **udp-log-on** option, above.

Syntax

```
sip ping server [any | udp | tcp | tls] [port]
sip ping-aor aor
sip traceroute server [any | udp | tcp | tls] [port]
sip server-monitor server [any | udp | tcp | tls] [port]
sip server-unload server [any | udp | tcp | tls] [port]
sip lookup-connection remoteIP [any | udp | tcp | tls] [localIP]
sip delete-connection remoteIP [any | udp | tcp | tls] [localIP]
sip purge-connection
sip reset-connection
sip clear-statistics
sip ping-resume
sip ping-suspend
sip udp-log-on
sip udp-log-off
```

Example

```
NNOS-E> sip ping 172.26.0.143
Sending OPTIONS to 172.26.0.143:5060 UDP
Success! Received OPTIONS Response 200:
  From: sip:172.26.0.153
  To: sip:172.26.0.143
```

srtp

Purpose

Provides a diagnostic tool for testing SRTP encryption/decryption of packets. When you select to decrypt a packet, use the same key and encryption suite that was used to originally encrypt the packet. When viewing the decrypt action output, if input and output are the same, the encryption is likely broken.

Select an action. The following fields are available for the **start** action.

- ***action***—specify whether to start, send, or stop the testing.
  - start—sets up the context for the testing session.
  - send—begins sending packets for testing.
  - stop—tears down the testing session.
- ***operation***—specify whether to encrypt or decrypt the specified packet.
- ***packetType***—select whether to send RTP or RTCP packets, as defined in RFC 3550, “RTP: A Transport Protocol for Real-Time Applications.”
- ***suite***—enter the SRTP protection suite consisting of encryption and authentication algorithms. Enter either:
  - None (no encryption or authentication)
  - AES-128 Countermode encryption, SHA-1 authentication (80 bit)
  - AES-128 Countermode encryption, SHA-1 authentication (32 bit)
  - F8-128 encryption, SHA-1 authentication
  - AES-128 Countermode encryption, MD5 authentication (Sipura)
  - DES Cipher Block Chaining per RFC-1889
- ***key***—enter the hexadecimal value of the master key/salt (it must include the “0x”) that is derived from the client or through AA-SBC tracing. This is the value typically passed in the SDP message, or the decrypted value of the key passed in Linksys INFO messages.
- ***mkiLen***—enter a value, in bytes, that sets the number of bytes in the master key identifier (MKI). Enter a value between 0 and 4. A value of 0 disables the MKI.
- ***mkiID***—enter the MKI identifier that should be included in each packet.

Actions

- ***roc***—enter value for the rollover counter. If it is a non-zero value, set the ssrc and sequence fields. This is state information used for testing.
- ***ssrc***—enter a value for the synchronization source. This is state information used for testing.
- ***sequence***—enter a sequence number. This is state information used for testing.
- ***flags***—enter a number, which will be used by AA-SBC, to complete the encryption/decryption testing.

The **send** action requires the packet field:

- ***packet***—enter the hexadecimal representation of the UDP payload of the packets (it must include the “0x”). You would typically obtain this with an Ethereal capture.

The **stop** action takes no parameters.

Syntax

```
srtp start {encrypt | decrypt} {rtp | rtcp} suite key [mki-len]
    [mki-id] [roc] [ssrc] [sequence] [flags]
srtp send packet
srtp stop
```

Example

The following example shows a successful SRTP decrypt:

```
NNOS-E> srtp start encrypt rtp AES_CM_128_HMAC_SHA1_80
    0x4b8730843d222bdc3ab023fb49b43b987e5611ccf4f4320db6bfa16eafc
NNOS-E> Success!
NNOS-E> srtp send
    0x8008000000123bad000000000000000000000000000000000000000000000000
    000000000000000000000000000000000000000000000000000000000000000000
    000000000000000000000000000000000000000000000000000000000000000000
    000000000000000000000000000000000000000000000000000000000000000000
    000000000000000000000000000000000000000000000000000000000000000000
    0000000000000000

SRTP encrypted packet (182 bytes):
    0x8008a60c00123bad000000000e183fbfb896e28b1848302c265db177a376deaa
    cade96d56dab52bc370f0ddff7bf9ec23d61d5bc9ecda0d781bafd62bcc61bab07
    0bf954829b8d02bddad2d75970d906b69cf764a98275fcd12968af8a4c43a1db3d
    985c3c5ba4b772079922315eac75efbf586a75e4f0e7eed615d549a034c6890dae
    3b5dbdf1fd3cef518891a9c7cdb5f0ed8cf0f119e137406e337838dfe17ce3c2cd
    eb340b4fb4a9764493b75e3056754e706d97
```

```
NNOS-E> srtp stop
AES_CM_128_HMAC_SHA1_80 Stats:   Pass             : 1   Encrypt
    : 1   Decrypt           : 0   Drop: Crypto       : 0   Drop: Auth
    : 0   Drop: Replay         : 0   Drop: Force          : 0   Drop:
    Internal       : 0
```

Actions

ssh

Purpose

Provides tools to manage SSH public keys. When installing a public key on to the system, make sure to select the OpenSSH format. You can Make sure

- **regenerate-host-key**—Regenerates the host keys used by SSH. Use this if you feel your SSH authentication system has been compromised. Note that AA-SBC shuts down and restarts when you execute this action.
- **import-public-key**—imports a public key into the AA-SBC key store. Use this option if the key already resides on the system as file.
- **add-public-key**—adds a public key to the key store. Use this option when you can paste or enter the key content in to the command line.
- **remove-public-key**—removes a public key from the key store. Use the **show ssh-public-keys** command to determine the number that corresponds with the key.
- **password**—assigns the SSH account password. use this option if you have set the **account** property of the ssh object to **ssh**.

Syntax

```
ssh regenerate-host-keys
ssh import-public-key fileName
ssh add-public-key key
ssh remove-public-key integer
ssh password password
```

Example

```
NNOS-E> ssh regenerate-host-keys
Are you sure (y or n)? y
Net-Net OS-E is restarting...

NNOS-E> ssh add-public-key "1023 37
   892894402808181734240272878525136869147515112964576516251364882012
   989536815157860119040084993502630627857986656332690235991583531077
   856659077009334439244103580273497657669693175159082203816786836588
   698494229203144203734802120585746029022983443556562727072942284125
   81920740438101310247843081310190326690376069 rsa-1"
Are you sure (y or n)? y
Success!
```

stest

Purpose

For Technical Support use only.

Syntax

```
stest options
```

terminal-failover

Purpose

Manages the database containing the active terminals known to AA-SBC. The database contains phone numbers and their associated devices, and is persistent across reboots. Select either:

- **merge**—merges the specified file into the active terminal list. Specify a file path; by default the system merges the file /cxc/terminal.xml.
- **replace**—replaces the active terminal list with the specified file. By default, the system replaces the list with the file /cxc/terminal.xml.
- **save**—saves the active terminal list cache to the database. By default, the system saves the cache to the file /cxc/terminal.xml. Optionally, you can specify a different file name.
- **delete**—deletes a specific entry from the from the active terminal list. Use the show active-terminals command to list entries.
- **flush**—removes all entries from the active terminal cache.

Syntax

```
terminal-failover merge [filePath]
terminal-failover replace [filePath]
terminal-failover save [filePath]
terminal-failover delete entry
terminal-failover flush
```

Example

```
NNOS-E> show active-terminals
```

Actions

```
address                     terminal                    type
-------                     --------                    ----
tel:+15086474840            SEP0013722AC21F             cisco
tel:+15086474850            SEP000E0C774E0B             cisco

NNOS-E> terminal-failover delete SEP000E0C774E0B
Success!
```

terminate-call

Purpose

Immediately disconnects the call associated with the specified session ID. This action is intended to be used only as a last resort when the endpoints are no longer reachable. It does not make any attempt to cancel the call at the endpoints (the UAs still believe that the call is in progress). Instead, it does an internal cleanup to remove the session from AA-SBC and free up any used resources. Use the **show active-call** command to view a list of currently active calls and their associated session IDs. Use the disconnect-call action to immediately terminate a call that has not been answered or that has been cancelled.

Syntax

```
terminate-call sessionID
```

Example

```
NNOS-E> terminate-call ca3e3764f07c42f5b7b6eda269d2d0c4
Success!
```

third-party-call-control

Purpose

Manages the database containing the active terminals known to AA-SBC. The database contains phone numbers and their associated devices, and is persistent across reboots. Select either:

- **call**—places a call.
- **hold**—places an existing call on hold.
- **retrieve**—retrieve an existing call from hold.
- **transfer**—transfer an existing call to another endpoint.
- **disconnect**—disconnect an existing call.
- **join**—given two calls, remove the originator from both calls and join the two terminators into a new call.
- **loop**—place a loopback call.
- **annotate**—attach an annotation on an existing call.
- **get-annotation**—display any annotation on an existing call.
- **park**—place a call to an endpoint, immediately placing it on hold.
- **connect**—place a call to an endpoint, and connect it to a parked call.
- **terminate**—disconnect one end of a call, leaving the other end on hold.
- **memo-begin**—start recording voice memo to a .wav file.
- **memo-end**—stop recording voice memo to a .wav file.
- **play**—play a .wav file on an existing call.
- **drop-file**— play a .wav file on an existing call, parking the originator.
- **notify**—send a NOTIFY event.
- **message**—connect to an endpoint, play a file, and terminate the call.

Syntax

```
third-party-call-control call to from [requestId] [enabled |
  disabled] [enabled | disabled] [any | UDP | TCP | TLS]
  [sessionConfigReference]
```

Actions

```
third-party-call-control hold handle [serverConfigReference]
third-party-call-control retrieve handle [serverConfigReference]
third-party-call-control transfer handle to [serverConfigReference]
third-party-call-control disconnect handle [serverConfigReference]
third-party-call-control join handle1 handle2 [serverConfigReference]
third-party-call-control loop handle [sessionConfigReference]
third-party-call-control annotate handle text [sessionConfigReference]
third-party-call-control get-annotation handle
   [sessionConfigReference]
third-party-call-control park endpoint [from] [requestId] [enabled |
   disabled] [sessionConfigReference] [serverConfigReference]
third-party-call-control connect handle endpoint [enabled | disabled]
   [requestId] [sessionConfigReference] [serverConfigReference]
third-party-call-control terminate handle [serverConfigReference]
third-party-call-control memo-begin handle filename [greeting]
   [enabled | disabled]
third-party-call-control memo-end handle [serverConfigReference]
third-party-call-control play handle filename [enabled | disabled]
   [serverConfigReference]
third-party-call-control drop-file handle filename
   [serverConfigReference]
third-party-call-control notify handle event [serverConfigReference]
third-party-call-control message filename endpoint [from] [requestId]
   [enabled | disabled] [sessionConfigReference]
   [serverConfigReference]
```

tls test

Purpose

Tests outgoing TLS connectivity to a remote device. Specify the destination IP address and port of the target device. Optionally, you can specify a reference to a configured certificate. If you do not specify the certificate, AA-SBC uses the default outgoing TLS certificate entry.

Syntax

```
tls test address:port [certificateReference]
```

Example

```
NNOS-E> tls test 10.1.80.2:5061 “vsp\tls\certificate Client”
Attempting TLS test connection.

Destination: 10.1.80.2:5061
Certificate: Record 129
```

```
TLS connection established; waiting for validation by remote TLS
   server...
TLS connection still up; connection must have passed validation.
   Success.!
```

If you do not specify a certificate and the remote peer required one, you would see this type of output:

```
NNOS-E> tls test 10.1.80.2:5061
Attempting TLS test connection.

Destination: 10.1.80.2:5061
Certificate: <Default Outgoing>

TLS connection lost: TLS handshake failure.
Connection attempt failed
```

If you specify the wrong certificate, you would see this type of output:

```
NNOS-E> tls test 10.1.80.2:5061 “vsp\tls\certificate Client”
Attempting TLS test connection.

Destination: 10.1.80.2:5061
Certificate: Record 129

TLS connection established; waiting for validation by remote TLS
   server...
TLS connection lost: Received TLS shutdown from peer.
Connection attempt failed
```

trap-reset

Purpose

Sends an acknowledgement to the SNMP agent to discontinue retransmission of SNMP traps. Use this action if you have enabled this feature with the **trap-retransmit** property of the snmp object. When you execute this action, all traps being retransmitted will be stopped. However, any subsequent traps will be retransmitted until you either re-execute this action or disable the **trap-retransmit** feature.

Syntax

```
trap-reset
```

Example

```
NNOS-E> trap-reset
Success!
```

umount

Purpose

Removes the specified device from the AA-SBC usable devices. In essence, this destroys a logical disconnection of the device, and you can no longer read or write to it. The effect of this is that any directories contained on the device become unavailable.

Syntax

```
unmount {data-1 | data-2 | usb | cdrom | system-1 | system-2}
```

Example

```
NNOS-E> umount data-2
Are you sure (y or n)? y
Success!
NNOS-E>
```

uri-alias

Purpose

Manages the alias table, an indexed table into the location database. This table maintains all the alias information for any known address of record. Select one of the following:

- **lookup**—returns all known aliases found in the alias table for the specified address of record.
- **reset**—deletes all alias mappings from the alias table and then recreates them by synchronizing with the enterprise directory. AA-SBC repopulates the table with any alias associated with a known AOR.

- **flush**—deletes all alias mappings from the alias table and repopulates the alias table as it relearns them. Optionally you can specify whether to flush the cache immediately or as the entries time out. By default, the system flushes entries immediately.
- **seek**—returns all known aliases found in the directory service for the specified address of record.
- **delete**—deletes the specified entry from the location alias table.
- **change-state**—changes the state of the location cache entry in the alias table. Enter the AOR and the new state.

You can also schedule this action as part of routine maintenance using the **task** object.

Syntax

```
uri-alias lookup aor
uri-alias reset
uri-alias flush [now | gracefully]
uri-alias seek aor
uri-alias delete aor
uri-alias change-state aor {unregistered | trying | in-service |
   redirect | registered | out-of-service}
```

Example

```
NNOS-E> uri-alias lookup sip:3000010004@tom.com
URL sip:3000010004@tom.com has the following aliases:
sip:3000010004@tom.com tag virtual
```

uri-resolve

Purpose

Returns the normalized URI and the address of record and binding for the specified URI. The URI can be a SIP URI (e.g., sip:joe@abc.com) or a TEL URI (e.g., tel:19788236666).

Syntax

```
uri-resolve uri
```

Example

```
NNOS-E> uri-resolve tel:+16667770001
URI tel:+16667770001 normalized to sip:6667770001@best.com
AOR sip:6667770001@best.com has a binding at 192.168.215.95 at 5060
   via UDP

NNOS-E> uri-resolve sip:6667770001@best.com
URI sip:6667770001@best.com normalized to sip:6667770001@best.com
AOR sip:6667770001@best.com has a binding at 192.168.215.95 at 5060
   via UDP
```

user-cache-lookup

Purpose

Performs a lookup in the AA-SBC user database for the requested AOR. The result returns the AOR with the associated UID and tag. You can use the **show user-cache** command to display all users in the cache. The AA-SBC directory process stores all entries for all configured enterprise directories in a database file. That file, which is read in to the SIP processing side or operations, creates a cache of all directory (user) information.

Syntax

```
user-cache-lookup aor
```

Example

```
NNOS-E> user-cache-lookup sip:jdoe@cov.com
Found - sip:jdoe@cov.com:4:test
```

vsp-reset

Purpose

Resets all sessions on the VSP, disconnecting any active sessions. This action is primarily a debugging tool and should only be used if Technical Supports instructs you to do so. However, it is also required to activate any changes to the properties in the static-stack-settings object.

Syntax

```
vsp-reset [vspName]
```

Example

```
NNOS-E> vsp-reset
Success!
NNOS-E>
```

web-services

Purpose

Sets the status/availability of a previously configured external location, policy, or event service server. Configure the services using the external-services group and service objects. Use the **show web-services-callout-details** command with the **availability** field to verify the status. Set the server status to one of the following:

- **disabled**—puts the server out of service until it is manually set to **available** using this action.
- **available**—returns a server to service with AA-SBC, whether it was manually set or detected as unavailable.
- **unavailable**—marks a server as temporarily unavailable. If a **heartbeat-url** is configured for the server, AA-SBC will attempt to bring it back into service.

Note that when AA-SBC detects that a server is unavailable, it automatically changes that web service server status to unavailable.If the status is unavailable, you must set it to available (once it is) with this action before AA-SBC can use that server again.

Syntax

```
web-services set-location server {disabled | available | unavailable}
web-services set-policy server {disabled | available | unavailable}
web-services set-event server {disabled | available | unavailable}
```

Example

```
NNOS-E> web-services set-policy “external-services policy-group pol1
   policy-service polSrvc1” available
No service for server reference
NNOS-E> config external-services policy-group pol1 policy-service
   polSrvc1
Creating ‘policy-group pol1’
Creating ‘policy-service polSrvc1’
config policy-service polSrvc1> exit
Do you want to commit your changes before you exit (y or n)? y
NNOS-E> web-services set-policy “external-services policy-group pol1
   policy-service polSrvc1” available
Success!>
```

xml

Purpose

Provides tools to manage XML files on AA-SBC. You must be running web services (enabled with the web object) on the system for this action to be available.

- **parse**—Checks the specified file for well formedness.
- **transform**—Translates the source file into the named destination file, using the specified style sheet. Enter a style sheet (.xsl file) to be used for the transformation. You can create your own to make modifications that you can then apply on each system. Optionally, you can use the **parameters** argument to alter the output transformed with the specified xsl style sheet.
- **validate**—Verifies that the content of the of the source file conforms to the default or specified schema. If you do not enter a schema, the system uses the configuration file as the default.

Syntax

```
xml parse file
xml transform stylesheet source destination [parameters]
xml validate fileName [schema]
```

Example

```
NNOS-E> xml parse /cxc/backup/cfg1106.xml
The XML file is not valid
NNOS-E> xml parse /cxc/install/install.xml
Success!
NNOS-E>

NNOS-E> xml transform serverUpdate.xsl /cxc/cxc.cfg /cxc/server.cfg
The XSL template appears invalid
NNOS-E> xml transform 2.1-to-3.0.xsl /cxc/backup/cxc.cfg /cxc/cxc.cfg
   red=blue
Success!
NNOS-E>

NNOS-E> xml validate /cxc/install.xml
Success!
```

4. Status provider show commands

Show command description

Global show command characteristics

All of the status show commands contained in this chapter share the same MIB and require the same user access level. In addition, they are available through a single place in the GUI.

Associated MIB

All status display commands reference the AA-SBC, Inc. enterprise MIB:

```
CXC.MIB
```

AA-SBC Management System path information

All status display commands are available in the AA-SBC Management System.

You can access the commands from the **Status** tab at the top of the window or from the **Status** link on the Home page. Expand the list on the left and click on a status report listed below it.

Filtering command output

The CLI allows you to filter output of show commands so that your display only includes the specific properties you requested. With no properties, **show** *object-name* displays all instances of the specified object. For example, if you execute the **actions** command, the CLI displays a list of all actions that have occurred in the current CLI session:

```
NNOS-E> show actions

action                    process timeout    requests   errors     timeouts
------                    ------- -------    --------   ------     --------
archive                     acct    10000       0          0          0
```

```
arp-delete              manager 10000      0          0          0
clock                   manager 10000      0          0          0
config merge            manager 30000      0          0          0
config replace          manager 30000      0          0          0
config save             manager 30000      2          0          0
database                manager 300000     0          0          0
database-maintenance    manager 10000      0          0          0
diameter                auth    10000      0          0          0
```

The output includes indices to the action (e.g., arp-delete, clock, config merge, config save, etc.) and properties of the indices (in this case, process, requests, errors, timeouts).

However, if you are only interested in seeing a specific index or property, you can filter on those fields. You can specify an index name to display only the instances with those values. Or, you can specify one or more property values to display only the instances with those property values.

Note: Index and property names are case *insensitive*.

To display a list of the properties you can filter on, enter the command with a question mark:

```
NNOS-E> show actions ?

 action provider statistics

 action
 process
 timeout
 requests
 errors
 timeouts
 -c        display the total number of instances
 -n        display a specified number of instances
 -v        verbose display
```

Filtering on an index

To filter on the index, enter the object name with the command (in quotation marks if the name includes white space). For example, you can display only the number of saves to the configuration file that have occurred:

```
NNOS-E> show actions “config save”

action              process timeout    requests   errors     timeouts
------              ------- -------    --------   ------     --------
config save         manager 30000      2          0          0
```

To filter on a property, enter the property name followed by an equal sign (or let the system enter the correct format with a TAB complete). To see only directory processes:

```
NNOS-E> show actions process=dir


action              process timeout    requests   errors     timeouts
------              ------- -------    --------   ------     --------
directory-reset         dir     120000     0          0          0
```

Note that you can enter multiple properties to further refine your output. You cannot, however, enter multiple instances of the same property. In that case, the last property entered is acted on:

```
NNOS-E> show actions process=SIP process=auth

action              process timeout    requests   errors     timeouts
------              ------- -------    --------   ------     --------
diameter            auth    10000      0          0          0
radius              auth    10000      0          0          0
```

Displaying total, count, and verbose reports

You can display summary reports on a status using one of the options defined in the following table. Following the table are examples of each.

Option	Description
`-c`	Displays a count of the total number of entries in a status report. Enter in the form: `show` *`object-name`* `-c`
`-n`	Displays the specified number of entries from the status report, counting from the first entry. Enter in the form: `show` *`object-name`* `-n=`*`x`* **-n=x** displays the first *x* instances.
`-v`	Displays a more detailed report of the object. This option does not change all output, only for those reports where summary and detailed reports are both available. Enter in the form: `show` *`object-name`* `-v`

Examples of -c, -n, and -v use

Without using display options, the output of **show dial-plan** looks like this:

```
NNOS-E> show dial-plan

plan-name     type     destination-url    from     peer-name   fwd
---------     ----     ---------------    ----     ---------   ---
abc.com       tag      aster              .*       abc         0
company.net   tag      bw                 .*       company     0
xyz.com       tag      aaa                .*       xyz         0
123.com       domain   sip:.*@123\.com    .*       123         0
server.com    domain   sip:.*@server\.com .*       server      0
```

Using the count option, the output of **show dial-plan -c** looks like this:

```
NNOS-E> show dial-plan -c

dial-plan returned 5 instances
```

Specifying that the system display the first two entries using the number of entries option, the output of **show dial-plan -n=2** looks like this:

```
NNOS-E> show dial-plan -n=2

plan-name      type     destination-url    from     peer-name    fwd
---------      ----     ---------------    ----     ---------    ---
abc.com        tag      aster              .*       abc          0
company.net    tag      bw                 .*       company      0
```

Using the verbose option, the output of **show dial-plan -v** looks like this:

```
NNOS-E> show dial-plan -v

                           plan-name: abc.com
                                type: tag
                                 url: aster
                     destination-url: aster
                              source: plan
                               level: 0
                                from: .*
                           peer-name: abc
                       peer-identity: sip:sametime@abc.com
                           peer-mode: provider
                              action: redirect
                                 fwd: 0
                                hits: 0
        incoming-host-normalizations:
                     in-request-user:
            in-request-user-template:
                          in-to-user:
                 in-to-user-template:
                        in-from-user:
               in-from-user-template:
        outgoing-host-normalizations:
--More--
```

Note: All show commands include the **-c**, **-n**, and **-v** options (although in some commands the options do not change output). Because they are universal, these options are not included in the command description syntax statement in this chapter.

show accounting-targets

Purpose

Displays information from all accounting targets configured on the NNOS-E. The settings are configured using the file-system object. See the file-client config object for information on the proper configuration when the external file-system is configured for SCP or SFTP.

Sample Output

```
NNOS-E> show accounting-targets
type: file-system
name: path 1
received: 0 CDRs
processed: 0 CDRs
failures: 0
missing-records: 0
average-processing-time: 0 milliseconds/CDR
```

Properties

Field	Description
type	The type of file-system target this command is displaying.
name	The name of the accounting target whose status is displayed.
received	The number of raw CDRs received.
processed	The number of CDRs processed.
failures	The number of failures.
missing-records	The number of raw CDRs the target found missing and could not write to the output. These messages may be missing or corrupt. A purge can cause this. Check logs for details.
average-processing-time	The average processing time per CDR of this accounting target.

show accounting-targets-file-system

Purpose

Displays information for each accounting target configured on the NNOS-E. This shows information for both file-system and external-file-system targets. See the file-client config object for information on the proper configuration when the external file-system is configured for SCP or SFTP.

There are four states that the external target cycles through as it processes raw CDRs, writes to the output file, and sends it to the remote system.

- Clear--The target is ready to write.
- Writing--The target is currently writing to the temporary file.
- Sending--The target is sending a file. At this time, the file can also be writing to a temporary file that will become the next file to send once the current file is successfully sent.
- Blocked--The target has one file in the middle of sending and another one ready to send. The target will not process anymore requests from the accounting server, but will send retries to the server giving retry interval based on its best estimate of when the retry can work.

Sample Output

```
NNOS-E> show accounting-targets-file-system
type: file-system
name: path 1
url:
master: enabled
state: clear
received: 0 CDRs
saved: 0 CDRs
files-sent: 0
current-file: /cxc_common/acct/test.2009.09.17.03.53.50.csv
cdrs-in-current-file: 0 CDRs
save-fails: 0
transmit-fails: 0
missing-records: 0
```

Properties

Field	Description
type	The type of file-system this command is displaying.
name	The name of the accounting target whose status is displayed.
url	The URL of this external accounting target.
master	Displays whether this box is a master for accounting master-service or not.
state	The state of the file system.
received	The number of raw CDRs received.
saved	The number of saved CDRs.
files-sent	The number of files sent to the target. This is applicable only to external file systems.
current-file	The file to which raw CDRs are currently being written. At the next rollover, this file is closed and a new one is opened.
cdrs-in-current-file	The number of CDRs in the current file.
save-fails	The number of failures that occurred during saving.
transmit-fails	The number of failures that occurred during transmission. This is applicable only to external file systems.
missing-records	The number of raw CDRs the target found missing and could not write to the output. These messages may be missing or corrupt. A purge can cause this. Check logs for details.

show active-session

Purpose

Displays message flow for a session. The message-log field indicates all messages for the session. In the first example, the session was a simple registration. In the second example, the session consisted of a call with multiple messages.

Sample Output

```
NNOS-E> show active-session
                     index: 1
                session-id: 0x4c2b67ad57e1cee
            association-id: 0x98000000006
             creation-time: 11:10:37.814301 Thu 2007-10-04
              session-type: proxy
            in-leg-call-id: bacd6b4a56394d5ea0f1fe0f6ae39e58
           out-leg-call-id: bacd6b4a56394d5ea0f1fe0f6ae39e58
         association-index: 3
               message-log: |-->INVITE|INVITE-->|INVITE 100<--|INVITE
   200<--|<--
INVITE 200|-->ACK|ACK-->|-->MESSAGE|MESSAGE-->|MESSAGE
   200<--|<--MESSAGE 200|-->
BYE|BYE-->|
ingress-classification-tag: qik-finemode
 egress-classification-tag: qik-finemode
```

Properties

Field	Description
index	A system-assigned internal identifier that indicates the current position of the session within the list of active sessions.
session-id	The internal identifier for the session. A session is a particular “conversation” between two endpoints.
association-id	The internal identifier for the association. An association is a pair of endpoints that might have had, be having, or in the future have, a “conversation.”
creation-time	A time-stamp indicating when the session was created.

Field	Description
session-type	The type of session being reported on, either: • **proxy**—stateful proxy • **stateless proxy**—stateless proxy • **b2bua**—B2B user agent • **outbound**—outbound call • **regServer**—registration server • **regClient**—registration client
in-leg-call-id	The call ID used for the call as it came into the system.
out-leg-call-id	The call ID the system used when forwarding the call out.
association-index	The index of the association that matches the From/To pair of this particular session.
message-log	A very short description of each message type that came through on the session.
ingress-classification-tag	The tag used to associate incoming traffic with the configured tag. The configured ingress-tag must match a configured ip routing-tag. You can also configure a classification-tag through the ip interface object. If this property is configured in both places, the session-config setting takes precedence.
egress-classification-tag	The tag used to select the outgoing interface. That tag must then be associated with an ip **routing-tag**, which controls the available egress interfaces and routes. You can also configure a **classification-tag** through the **ip interface** object. If this property is configured in both places, the **session-config** setting takes precedence.

show authentication-details

Purpose

Displays authentication error details on the AA-SBC.

Sample Output

```
NNOS-E>show authentication-details
-------------------------------------------------------------------------------
Provider      Requests   Replies  Accepts  Rejects  Timeouts  QClipped   Others
-------------------------------------------------------------------------------
Local                0         0        0        0         0         0        0
RADIUS               3         3        0        0         3         0        0
Diameter             0         0        0        0         0         0        0
Directory            0         0        0        0         0         0        0
Accept               0         0        0        0         0         0        0
Reject               0         0        0        0         0         0        0
-------------------------------------------------------------------------------
```

Properties

Field	Description
Provider	The protocol to be used for errors.
Requests	The number of requests submitted to each provider.
Replies	The number of replies to errors.
Accepts	The number of positive replies received from the remote server.
Rejects	The number of rejects received from the remote server.
Timeouts	The number of timeouts that have caused errors.

Field	Description
QClipped	The number of errors that have failed locally, without ever being sent to the remote server because the queue of requests outstanding to the server(s) has grown too long.
Others	Sum of other errors. These can be seen individually by adding **-v** to the end of the action.

show automatic-settings

Purpose

Displays values that AA-SBC has generated for each property that supports the automatic settings feature. For these properties, AA-SBC determines an appropriate value based on various aspects of the system hardware, such as the platform, CPU speed, and available memory.

Note: Do not change the values of properties configured with automatic-settings unless instructed to do so by Technical Support.

Sample Output

```
NNOS-E> show automatic-settings

name                          value
----                          -----
cac-max-calls                 7500
cac-max-calls-in-setup        1500
cac-max-number-of-tls         3000
cac-max-tls-in-setup          423
cac-min-calls-in-setup        10
max-number-of-sessions        7500
stack-socket-event-threads-max 4
stack-socket-threads-max      4
stack-worker-threads          4
```

Properties

Field	Description
name	The name of the property whose default value is automatically determined by the system. For example, **cac-max-calls** sets the maximum number of concurrent calls allowed on the VSP.
value	The default value that the system assigns to the property.

show boxes

Purpose

Displays information for all boxes configured within AA-SBC. Configure boxes using the box object.

Sample Output

```
NNOS-E> show boxes

------------------------------------------------------------------------------
Box Address       ? Prot  State        Up Time    Connects  Errors  Last Error
------------------------------------------------------------------------------
Local             O None  Connected    01:09:20          1       0  Unknown
192.168.0.2       A TCP   Connected    01:09:10          1       0  None
------------------------------------------------------------------------------
```

Purpose

Field	Description
Box Address	The IP address for all boxes in the cluster (or Local to indicate the local box).
?	The role of the box in its connection to the local box, either originator (O) or accepter (A).

Field	Description
Prot	The messaging protocol in use between boxes, either TCP or TLS.
State	The state of the connection between boxes, either: • Idle • Connecting • Helloing • Connected • Waiting (the box is not connected and AA-SBC is waiting a short period before attempting to reconnect).
Up Time	If **State** is **Connected**, the time of connection between boxes.
Connects	The number of times the system has successfully connected to the local box since 1) this AA-SBC booted, and 2) the device was added to the cluster.
Errors	The number of attempts to reconnect that were unsuccessful.
Last Error	The type of the last error. Error types are: • **None**—no error. • **No Route**—No route can be found to remote box. • **No Socket**—Failed to create socket to connect to this box. • **No Connect**—Connection failed (e.g., due to box is down, or network or configuration error). • **Connect Timeout**—Connection timed out. • **Disconnect**—Boxes were disconnected. • **Loopback**—Configuration error, i.e., the “remote” box is the local box. • **Duplicate MAC**—Duplicate MAC address detected (probably loopback). • **Hello Timeout**—Connected, but failed to communicate. • **Version Mismatch**— Communicated, but discovered incompatible versions. • **Keepalive Failed**—Connected, but box didn't respond to keepalive messages. • **Other**— Other type of error.

Status provider show commands

show call-admission-control

Purpose

Displays settings and statistics for call admission control on this VSP (INVITE requests). The name field identifies the VSP being reported on. The settings are configured using the admission-control object.

Sample Output

```
NNOS-E> show call-admission-control
                                    name: default
                  call-admission-control: enabled
                               max-calls: 7500
                      max-calls-in-setup: 1500
                      min-calls-in-setup: 10
        calls-in-setup-dynamic-threshold: 1500
                        cpu-monitor-span: 20 seconds
                    cpu-monitor-interval: 10 seconds
                         average-sip-cpu: 0 %
                calls-high-cpu-threshold: 90 %
                 calls-low-cpu-threshold: 50 %
                           current-calls: 0
                  current-calls-in-setup: 0
                              most-calls: 0
                     most-calls-in-setup: 0
                       max-calls-dropped: 0
          max-calls-dropped-last-logging:
max-calls-in-setup-dropped-this-interval: 0
max-calls-in-setup-dropped-last-interval: 0
              max-calls-in-setup-dropped: 0
```

Properties

Field	Description
name	The name of the VSP whose status is displayed.
call-admission-control	The state of CAC for this VSP—whether it is enabled or disabled.
max-calls	The maximum number of calls allowed on this VSP. This is the overall simultaneous call limit.

Field	Description
max-calls-in-setup	The maximum number of simultaneous inbound and outbound call legs in the setup stage allowed by the CAC.
min-calls-in-setup	The minimum number of simultaneous inbound and outbound call legs in the setup stage allowed by the CAC.
calls-in-setup-dynamic-threshold	The limit for the number of in-progress calls allowed before the system suppresses all calls.
cpu-monitor-span	The number of seconds over which the system calculates the total system CPU average. At the conclusion of the span, the average value is compared to the call and registration CPU thresholds to determine whether to modify the dynamic threshold. The longer the span, the fewer the changes to the thresholds. A shorter span will result in reactions to brief CPU activity spikes.
cpu-monitor-interval	The frequency, in seconds, with which the system calculates the total system CPU average for the last span.
average-sip-cpu	The current average CPU usage.
calls-high-cpu-threshold	The percentage value of CPU usage that determines whether the system modifies the call dynamic threshold.
calls-low-cpu-threshold	The lowest percentage value of CPU usage that the system can drop to when decreasing the dynamic threshold. The system starts decreasing the dynamic threshold when the average CPU usage exceeds the value for calls-cpu-threshold.
current-calls	The number of calls currently being processed by the system.
current-calls-in-setup	The number of calls currently in the setup stage on the system.
most-calls	The highest number of calls processed at any one time (since last system boot).
most-calls-in-setup	The highest number of calls in setup stage at any one time (since last system boot).

Status provider show commands

Field	Description
max-calls-dropped	The total number of active calls that were dropped since the last system boot.
max-calls-in-setup-dropped-this-interval	The number of calls that were in setup stage but dropped during the current interval. The interval is defined with the **cpu-monitor-interval** property.
max-calls-in-setup-dropped-last-interval	The number of calls that were in setup stage but dropped during the previous interval. The interval is defined with the **cpu-monitor-interval** property.
max-calls-in-setup-dropped	The maximum number of calls in the setup stage that were dropped since the last system boot.

show call-routing

Purpose

Displays the call routing table, which defines how AA-SBC forwards an outgoing call. The output displays a summary of each *active* dial-plan entry, its match criteria and peer (and other configuration elements), and the number of times AA-SBC has applied it to forward a call. Use the show dial-plan command to see all *configured* dial plans.

Sample Output

```
NNOS-E> show call-routing
----------------------------------------------------------------------
Forwards  Pri  Type    Data
----------------------------------------------------------------------
       5  100  domain  Plan name:  companyXYZ.com
                       Match:      companyXYZ.com
                       Peer name: Company ST
```

Properties

Field	Description
forwards	The number of times this plan has matched an INVITE request, and the system forwarded the request. This is a counter internal to the system.
pri	The priority (order of preference) setting for the dial-plan entry. This property overrides the default behavior (most specific match) and sets a preference based on the **request-uri-match** (route) or **source-match** (source-route) property.
type	The portion of the request to match on. If the INVITE matches the portion identified by the type, the system forwards the request to that server. The type can be contributed from the dial-plan configuration. Types of *tag* or *domain* can be contributed from the **auto-tag-match** and **auto-domain-match** options of the **server routing-setting** property.
Data	The data field is made up of Plan name, Match, an Peer name, all of which are described below.
Plan name	The name of the active dial plan, created with the dial-plan or dial-prefix configuration.
Match	A derivative of the regular expression or tag (for faster matching) that identifies the "to" or "from" mapping. This string is configured in the dial-plan configuration. If contributed through a route object entry, the string to match in the SIP header fields or transport information in order for the system to apply the plan to calls containing the prefix ("to" mapping). If contributed through a source-route object entry, the match criteria for the source of the SIP message ("from" mapping). The **match** field is derived from either the **to-uri-match** (route) or **source-match** (source-route) property. If type is condition-list, the match is derived from the priority plus plan name.
Peer name	A statically entered peer. This is a configured **server** of type **sip-registrar**.

show chassis-info

Purpose

Displays hardware and firmware information (e.g. serial, part, and version numbers) for the chassis. The three fields described are useful information for Technical Support. All other fields are Intel-specific, and are not used at this time.

Sample Output

```
NNOS-E> show chassis-info
             BIOS-version: 6.7.1.1
          chassis-version: 1
             chassis-type: 17
      chassis-part-number: NN 2610
    chassis-serial-number: 201-01060
           chassis-custom: SR1500
            board-version: 1
          board-lang-code: 25
           board-mfg-time: 16:30:00 Sun 2026-09-27
       board-manufacturer: Intel
       board-product-name: SE7520JR22
      board-serial-number: BZJR44325226
        board-part-number: C53660-403
        board-fru-file-id: FRU Ver 0.01
             board-custom:
          product-version: 1
        product-lang-code: 25
     product-manufacturer: Intel
             product-name:
product-part-model-number:
         product-revision:
    product-serial-number:
        product-asset-tag: 1234
      product-fru-file-id:
           product-custom:
```

Properties

Field	Description
BIOS-version	The current software update package revision; the revision of firmware that the AA-SBC hardware is running. Also sometimes known as the System Update Package (SUP), it controls fans, power, IMM (Management Module), BIOS, and more.
chassis-part-number	The model number of the current device.
chassis-serial-number	The associated part number of the current AA-SBC device.

show clock

Purpose

Displays the current date and time, and the amount of uptime in days, hours, and minutes since AA-SBC was started. You can set the time with the clock action.

Sample Output

```
NNOS-E> show clock
  time: 12:05:32 Thu 2006-12-21
uptime: 0 days 01:09:20
```

Properties

Field	Description
time	The current time as configured on the box. You can set (or reset) the system time with the **clock** action
uptime	The amount of time since the last system boot.

show cluster

Purpose

Displays each box, by IP address, that is part of the cluster. Additionally, the output indicates whether the box is receiving, or due to receive, configuration from the cluster master. The output displays the configuration of the **cluster** and **box** objects.

Sample Output

```
NNOS-E> show cluster
ip-address       box-id
----------       ------
0.0.0.0          1
192.168.0.2      2
```

Properties

Field	Description
ip-address	The IP address of each box connected to this local box. The local box displays as 0.0.0.0.
box-id	The ID assigned to the box. This is the identifier assigned through the box object.

show collect-status-classes

Purpose

The **show collect-status-classes** action displays which status classes are being collected. When entered with the **default** parameter, the AA-SBC default status classes are listed.

You can also use the **show collect-status-classes** status provider to display status classes defined in custom configurations.

Sample Output

```
NNOS-E>show collect-status-classes accounting

Status classes to be collected for 'Accounting':
-------------------------------------------------------------------------------
Source   Status class                Description
-------------------------------------------------------------------------------
config   accounting-recent           calls recently accounted
config   accounting-database         request information for accounting database
connections
config   accounting-files            accounting file information
config   accounting-store            accounting disk storage information
config   accounting-cdr-summary      accounting CDR summary
config   accounting-targets-file-system accounting file-system and
external-file-system targets
config   accounting-targets          accounting targets
```

show cpu-usage

Purpose

Displays CPU usage over various preset time intervals. Use the cpu-monitor action to do live monitoring of system use.

Sample Output

```
NNOS-E> show cpu-usage
 1 second: 0 %
10 second: 0 %
 1 minute: 1 %
10 minute: 2 %
   1 hour: 7 %
```

Properties

Field	Description
1 second	The last reading of CPU usage on the system.
10 seconds	The average CPU usage on the system for the last 10 seconds.
1 minute	The average CPU usage on the system for the last one minute.
10 minutes	The average CPU usage on the system for the last 10 minutes.
1 hour	The average CPU usage on the system for the last one hour.

show database-maintenance-status

Purpose

Displays the current maintenance status of database operations. Use this to determine whether an operation (e.g., a backup or restore) has finished. Or, if you receive an error that AA-SBC could not execute a database operation, check this status to verify the state of the database. All previous operations must be complete (indicated by a status of idle) before a new operation can begin.

Sample Output

```
NNOS-E> show database-maintenance-status
  status: backup
   table:
 started: 09:51:23 Fri 2007-10-05
finished: 09:51:25 Fri 2007-10-05
  result: Success!
```

Properties

Field	Description
status	The current progress of database maintenance operations (initiated by either the **database-backup** or **database-maintenance** action or task). You can see the list of database tables using the **show database-tables** command. Table names are listed in parenthesis next to the descriptions below. The state reported indicates that the system is: • **idle**—there are no current operations; you can initiate an action. • **upgrading**—executing a set of database upgrade commands to upgrade the database to a new version. The upgrade process compares the existing database version on the box to the upgrade package, and upgrades if the package version is newer. • **initializing**—initializing the database and loading the stored procedure calls. continued

Field	Description
maintenance-status *continued*	• **reindexing**—re-indexing the database tables. • **analyzing**— collecting statistics about the contents of tables in the database. • **purging-sip**— deleting SIP message table (sipmessage) entries. • **purging-transport**—deleting transport table (spotlitetransportmsg) entries. • **purging-RTCP-Tx**—deleting RTCP transmit table (spotlitertcptxmsg) entries. • **purging-RTCP-Rx**—deleting RTCP receive table (spotlitertcprxmsg) entries. • **purging-monitored-calls**—deleting monitored calls (monitoredcalls) table. • **purging-URL**—deleting URL table (urlmsg) entries. • **purging-acct**—deleting accounting table (acctcallstruct) entries. • **purging-media**—purging media message (mediamsg) table entries. • **purging-file-transfer**—purging file transfer (filetransmsg) message table entries. • **purging-archive-IM**—purging archived IM messages (archiveimmsg) table entries. • **purging-calllegstart**—purging call-leg-start (calllegstart) table entries. • **purging-calllegstop**—purging call-leg-stop (calllegstop) table entries. • **vacuuming**—reclaiming storage occupied by deleted entries. • **backup**—performing a pg_dump operation. • **restore**—restoring the database from a previously backed up version. • **failed**—operation failed. Retry the operation or reload the database before calling Technical Support. • **translating-tables**—translating data into the current database format to allow for better database query performance. This state only appears when upgrading a system from 3.2.0 or earlier to a later release.
table	The name of the database table.
started	The time at which the database maintenance operation was started.

Status provider show commands

Field	Description
finished	The time at which the database maintenance operation was completed.
result	An indication of whether the database maintenance operation was successful or not.

show dial-plan

Purpose

Displays the dial-plan table, which handles call forwarding. The output displays a summary of each *configured* (but not necessarily active) dial-plan entry, its match criteria and peer (and other configuration elements), and the number of times AA-SBC has applied it to forward a call. Use the show call-routing command to see all *active* dial plans.

Sample Output

```
NNOS-E> show dial-plan
plan-name      type      match       min     pri     peer-name     fwd
---------      ----      -----       ---     ---     ---------     ---
E911           default   !*          2      99                    0
default        phone     !*          2      100     Verizon       0
New York       phone    212!*       3      100     NNOS-E@NewYork   0
San Jose       phone    506!*       3      100     NNOS-E@SanJose   0
Boston         phone     617!*       1      100                   0
Maynard        phone     978!*       1      100                   0
```

Properties

Field	Description
plan-name	The name of the active dial plan, created with the dial-plan or dial-prefix configuration.
type	The portion of the request to match on. If the INVITE matches the portion identified by the type, the system forwards the request to that server. The type can be contributed from the dial-plan configuration. Types of *tag* or *domain* can be contributed from the **auto-tag-match** and **auto-domain-match** options of the server **routing-setting** property.
match	A derivative of the regular expression or tag (for faster matching) that identifies the “to” or “from” mapping. This string is configured in the dial-plan configuration. If contributed through a **route** object entry, the string to match in the SIP header fields or transport information in order for the system to apply the plan to calls containing the prefix (“to” mapping). If contributed through a **source-route** object entry, the match criteria for the source of the SIP message (“from” mapping). The **match** field is derived from either the **to-uri-match** (route) or **source-match** (source-route) property. If type is condition-list, the match is derived from the priority plus plan name.
min	The minimum number of digits required for a match on a phone prefix, if configured. In some cases, the system calculates a value for other types of matches based on the number of characters (including wild cards). In some cases it displays as-is. The value is only meaningful to a phone-prefix match, however.
pri	The priority (order of preference) setting for the dial-plan entry. This property overrides the default behavior (most specific match) and sets a preference based on the **request-uri-match** (route) or **source-match** (source-route) property.

Field	Description
peer-name	A statically entered peer. This is a configured server of type **sip-registrar**.
fwd	The number of times this plan has matched an INVITE request, and the system forwarded the request. This is a counter internal to AA-SBC.

show dns-cache

Purpose

Displays the DNS cache, organized by process (monitor, manager, SIP, media, auth, etc.), on AA-SBC. The cache displays host information, including type, state, and references. Configure DNS using the dns object.

Sample Output

```
NNOS-E> show dns-cache
process   name            type    ttl          state        references
-------   ----            ----    ---          -----        ----------
manager   127.0.0.1       PTR     static       Resolved     0
manager   172.26.0.155    A       static       Resolved     0
manager   192.168.0.1     PTR     static       Resolved     0
SIP       10.1.34.160     PTR     static       Resolved     0
SIP       172.26.0.155    A       static       Resolved     0
SIP       192.168.0.1     PTR     static       Resolved     0
SIP       localhost       A       static       Resolved     0
SIP       vfn.com         NS      169023       Resolved     0
media     10.1.34.160     PTR     static       Resolved     0
media     127.0.0.1       PTR     static       Resolved     0
media     localhost       A       static       Resolved     0
reg       10.1.34.160     PTR     static       Resolved     0
reg       127.0.0.1       PTR     static       Resolved     0
reg       172.26.0.155    A       static       Resolved     0
reg       localhost       A       static       Resolved     0
```

Properties

Field	Description
process	The name of the system process that did the DNS cache lookup for an entry. That entry is then installed in the process cache (each process has its own cache). A static entry is installed in every process cache.
name	The identifier for the entry (e.g., IP address, host name, etc.). The name format is determined by the record type.

Field	Description
type	The record type for the entry, either: • **A**—host name is a IPv4 address • **AAAA**—host name is a IPv6 address • **PTR**—IP address is an address-to name-mapping pointer record (reverse lookup) • **SRV**—service name (server resource rule) • **NAPTR**—domain name (Naming Authority Pointer rule) • **CNAME**—canonical name record (makes one domain name an alias of another) • **NS**—name server record • **SOA**—server of authority record
ttl	The time to live for the entry, either a number of milliseconds or static. A value of static indicates that the entry was manually entered and will not time out of the cache.
state	The state of the entry in the cache, either Pending (resolution in progress), Resolved, or Not Available.
references	The number of accesses to that cache entry.

show dns-cache-detail

Purpose

Displays DNS cache entries, organized by process. An entry is installed in a process cache (each process has its own cache) when the process does a DNS lookup for that entry. A static entry, configured with the dns object, is installed in every process cache.

Sample Output

```
NNOS-E> show dns-cache-detail
Process: manager
----------------------------------------------------------------------
DNS name            type    ttl      data
----------------------------------------------------------------------
10.1.34.160         PTR     static   lingo.com
127.0.0.1           PTR     static   localhost
172.26.0.155        A       static   192.168.0.1
```

```
192.168.0.1          PTR     static   172.26.0.155
localhost            A       static   127.0.0.1

Process: SIP
----------------------------------------------------------------------
DNS name             type    ttl      data
----------------------------------------------------------------------
10.1.34.160          PTR     static   lingo.com
127.0.0.1            PTR     static   localhost
172.26.0.155         A       static   192.168.0.1
192.168.0.1          PTR     static   172.26.0.155
lingo.com            A       static   10.1.34.160
localhost            A       static   127.0.0.1
```

Properties

Field	Description
DNS name	The main name used by AA-SBC for the DNS entry. This is the name configured as the **host**, **service**, or **naptr** in the **dns** configuration object (static record) or learned by AA-SBC.
type	The record type for the entry, either: • **A**—host name is a IPv4 address • **AAAA**—host name is a IPv6 address • **PTR**—IP address is an address-to name-mapping pointer record • **SRV**—service name (server resource rule) • **NAPTR**—domain name (Naming Authority Pointer rule) • **CNAME**—canonical name record (makes one domain name an alias of another) • **NS**—name server record • **SOA**—server of authority record

Field	Description
ttl	The time to live for the entry, either a number of milliseconds or static. A value of static indicates that the entry was manually entered and will not time out of the cache.
Data	The resolution of the DNS lookup, depending on the record type. For PTR and A records, this field contains an IP address or domain name. For SRV and NAPTR records, the data field displays the configured rules that describe lookup procedures and precedence.

show dns-resolver

Purpose

Displays information identifying the servers that receive AA-SBC resolver requests and various counters. As a resolver, AA-SBC obtains resource records from servers on behalf of resident or requesting applications. The load-balancing algorithms determine which server receives the request. Note that if you change the DNS server configuration (with the **server** property of the resolver object), AA-SBC resets the DNS resolver statistics.

Sample Output

```
NNOS-E> show dns-resolver
        process: manager
     preference: 100
         server: 172.31.4.50:53
     domainName: vfn.com
       protocol: UDP
          state: up
   echoRequests: 0
    echoReplies: 0
       requests: 0
      responses: 0
  request-fails: 1
       discards: 0
        retries: 1
pending-queries: 1
   sip-location: ALL

        process: manager
     preference: 100
         server: 172.31.4.51:53
     domainName: vfn.com
       protocol: UDP
          state: up
   echoRequests: 0
    echoReplies: 0
       requests: 0
      responses: 0
  request-fails: 0
       discards: 0
        retries: 0
pending-queries: 4294967295
   sip-location: ALL

--more--
```

Properties

Field	Description
process	Sorted by system process, the DNS resolver recognized by that process.
preference	The preference assigned to the server, configured with the **server** property of the **resolver** object. This value is used if the **server-scheme** property, which specifies how the system selects the server for forwarding DNS queries, is set to **preference-order**.
server	The IP address of the server used for DNS queries, either the system dns-server or an external server.
domainName	The domain name mapped to the server IP address, via the static-stack-settings **domain-name** property.
protocol	The protocol used by the server, either any, UDP, TCP, or TLS.

Field	Description
state	The state of the DNS server, either up or down. If the resolver **failure-detection** property is set to **enabled**, and the server is down, the system clears the DNS cache, resets DNS sockets, and instructs servers of type dns-group to clean up states. If the property is **disabled** and the server is down, the system caches the entry as not available.
echoRequests	The number of ICMP echo requests sent to the DNS server.
echoReplies	The number of ICMP echo responses sent from the server.
requests	The number of DNS requests sent to the server.
responses	The number of responses to DNS requests sent from the server.
request-fails	The number of transaction-layer failures (e.g., timeout or socket failure).
discards	The number of discarded resource record requests.
retries	The number of retries on request failures.

Field	Description
pending-queries	The number of as-yet unanswered queries awaiting resource record responses from the DNS server.
sip-location	The DNS lookup behavior, which determines how the system should attempt to locate a SIP server using DNS when it receives a SIP message that is not destined to any dial plan or locally registered user. If configured as a resolver, the system attempts to obtain the SIP NAPTR or SRV record for the domain. This parameter sets the behavior if the system does not find that NAPTR or SRV record. Method could be: • ALL—uses NAPTR, SRV or Address records (the default). • NONE—prevents the use of DNS to locate a SIP server. • SRV—uses the service record only (no NAPTR lookup). • AAA—uses the Address record only (no NAPTR/SRV lookup). • RFC3263—uses NAPTR and SRV records as described in *RFC 3263, Session Initiation Protocol (SIP): Locating SIP Servers* (no AAA lookup).

show dos-rules

Purpose

Displays a summary and statistics for each DOS rule (transport and SIP) created by your DOS policies. It displays the results of each rule generated by the DOS engine (what got locked out). The command only displays output if:

- AA-SBC is currently under DOS attack.
- You have configured DOS policies that are thwarting that attack.

Sample Output

```
NNOS-E> show dos-rule
rule-number: 1
```

```
last-timestamp: 13:22:17 Wed 2007-04-11
packet-count: 610825
action: filter
missive: sip policy: s1, filter localIP: 10.1.208.140 requestMethod:
   INVITE requestUriUserHost: proxy.companyXYZ.com
```

Properties

Field	Description
rule-number	Number assigned to rule in the index of rules.
last-timestamp	The last time a packet tripped the rule.
packet-count	Total number of packets to hit this rule.
action	The action to take. Either: • Filter these SIP packets (throw away). • Alert (allow these SIP packets to go through, but send a log message and SNMP traps).
missive	A statement for the given policy rule specifying what kind of packets to look for and which ones to drop.

show ethernet

Purpose

Displays configuration and status information for each configured Ethernet interface.

Sample Output

```
NNOS-E> show ethernet
name  link speed duplex autoneg
----  ---- ----- ------ -------
eth0  up   1Gb   full   enabled
eth1  up   1Gb   full   enabled
eth2  up   1Gb   full   enabled
eth3  down              enabled
```

Properties

Field	Description
name	The name of the Ethernet interface. You can configure up to 20 gigabit Ethernet, full-duplex interfaces. The actual number available depends on your hardware configuration
link	The operational state of the Ethernet interface, either up or down. For the link to be up, the interface must be administratively enabled with a link detected on the line.
speed	The speed of the Ethernet connection between the NNOS-E and the piece of equipment to which it is connected. The value displayed is the speed configured with the **speed** property of the **interface** object. However, AA-SBC ignores this value if **autoneg** is set to **enabled**.
duplex	The acceptable duplex method for the interface, either half (asynchronous) or full (simultaneous) transmission. The value displayed is the setting configured with the **duplex** property of the **interface** object. However, AA-SBC ignores this value if **autoneg** is set to **enabled**.
autoneg	The autonegotiation setting for the interface. If **enabled**, the system negotiates with the piece of equipment to which it is connected to achieve optimal agreed upon settings. If **disabled**, the system uses the configured (or default) settings.

show event-log

Purpose

Displays the time, severity, box, process, and message for each event in the AA-SBC local database. You must enable global event-log administration (through the event-log object) for data to be written to the event log. Set the severity level, log class, and history properties using the event-log local-database object. See the appendix for a list of events and their corresponding severities.

Sample Output

```
NNOS-E> show event-log
timestamp               severity   box      process   class      mess
   age
---------               --------   ---      -------   -----      ----
   ---
10:57:27 Tue 2006-11-07  error      1        manager   sm         Cert
   entry 'vfn' could not parse certificate file '/cxc/certs/vfn.p12';
   could be wrong passphrase (secret tag is pass), or unsupported
   format (PEM or PKCS#12)
10:57:28 Tue
   2006-11-07  error      1        manager   sm         OpenSSL error
   from PKCS12_parse()() (returned 0):
10:57:33 Tue
   2006-11-07  error      1        SIP       sm         OpenSSL error
   23076071: error:23076071:PKCS12 routines:PKCS12_parse:mac verify
   failure (p12_kiss.c:117)
10:57:33 Tue
   2006-11-07  alert      1        SIP       sipTLS     Reinitializin
   g table AssociationSQL in database spotlite
10:57:40 Tue 2006-11-07  alert      1        SIP       snmp       VRRP
   Group 2 failover
10:57:46 Tue 2006-11-07  alert      1        manager   snmp       VRRP
   vinterface vx112 failover: went to master on interface eth2
```

Properties

Field	Description
timestamp	The time at which the event occurred, according to the system clock.
severity	The severity of the event. The severity indicates the lowest level message to display. You receive messages of that class and below, with Emergency being the lowest and Debug the highest. Set the severity level you wish to display using the local-database object **filter** property.
box	The box in the cluster on which the event occurred. If the box is not in a cluster, the system reports the box number as 1.
process	The process that sent the event to the local database. You can display a list of eligible processes using the show processes command.
class	The log class related to the event, which indicates the subsystem that generated the message. Set the severity level you wish to display using the local-database object **filter** property.
message	The text of the event message.

show faults

Purpose

Displays crash dump data, including time, version, file, and location information.

Sample Output

```
NNOS-E> show faults
   time: 16:32:53 Wed 2006-12-20
   file: SIP1-000.txt
address:
 reason: Aborted
 uptime: 1 days 00:10:15
version: 3.2.0
  build: 22831
 branch: b3.2.0

   time: 16:41:48 Wed 2006-12-20
   file: SIP1-001.txt
address:
 reason: Aborted
 uptime: 0 days 00:07:52
version: 3.2.0
  build: 22831
 branch: b3.2.0

   time: 17:37:15 Wed 2006-12-20
   file: SIP1-002.txt
address: b7cfbc04 __vsnprintf + 0x59 /lib/libc-2.3.4.so
 reason: Segmentation fault
 uptime: 0 days 00:55:20
version: 3.2.0
  build: 22831
 branch: b3.2.0
```

Properties

Field	Description
time	System timestamp indicating the time of the crash.
file	The name of the file to which the system wrote the crash data and analysis. Files are stored in the directory cxc_common/crash.
address	The specific point at which the crash occurred. The **file** (above) contains a complete backtrace of the crash.
reason	A brief description of the cause of the crash, for example, OOM (out of memory) or segmentation fault.
uptime	The length of time the system was up before the crash occurred.
version	The software version installed on the system at the time of the crash.
build	The specific build of the software version that was installed on the system at the time of the crash.
branch	The internal development tracking ID for the build that was installed on the system at the time of the crash.

show features

Purpose

Displays each licensed feature for AA-SBC and its capacity. The output also displays any changes to capacity (either number of sessions or endpoints), current use, and a running total of use. Capacity changes can be implemented using the features object. The example below shows only a small portion of the output. Fields displayed are the same for each feature type. See the **features** object description in Chapter 30, “Features licensing objects” for a description of each licensable feature.

Sample Output

```
NNOS-E> show features
 feature: signaling-sessions
licensed: 200000
 current: 1
 maximum: 6
   total: 88
failures: 0

 feature: media-sessions
licensed: 200000
 current: 1
 maximum: 0
   total: 9
failures: 0

 feature: instant-message-and-presence-sessions
licensed: 200000
 current: 0
 maximum: 0
   total: 0
failures: 0
.
.
.
```

Properties

Field	Description
feature	The name of the feature that is licensed. If a feature is not licensed, it does not display in this list.
licensed	The number of sessions or endpoints allowed concurrently, as specified in the license. Typically this is the value from the license, but if you reset the value with the features object, that setting displays here.
current	The number of sessions or endpoints using the license at the moment of command execution.
maximum	The maximum number of licenses ever used on the box.
total	The total number of accesses to the feature.
failures	The number of times, after a license maximum was reached, that a user or application tried to use the feature.

show interface-details

Purpose

Displays statistics and the MAC address for each configured Ethernet interface. Interfaces are configured using the interface object.

Sample Output

```
NNOS-E> show interface-details
          name: eth0
   mac-address: 00:04:23:c3:22:04
      op-state: up
     rcv-bytes: 27723136
   rcv-packets: 195105
      rcv-errs: 0
      rcv-drop: 0
      rcv-fifo: 0
     rcv-frame: 0
rcv-compressed: 0
```

```
         rcv-multicast: 0
              tx-bytes: 220274389
            tx-packets: 3350328
               tx-errs: 0
               tx-drop: 0
               tx-fifo: 0
              tx-colls: 0
            tx-carrier: 0
         tx-compressed: 0

                  name: eth1
           mac-address: 00:04:23:c3:22:05
              op-state: up
             rcv-bytes: 1978425176
           rcv-packets: 6394797
              rcv-errs: 0
              rcv-drop: 0
              rcv-fifo: 0
             rcv-frame: 0
        rcv-compressed: 0
         rcv-multicast: 0
              tx-bytes: 1335807214
            tx-packets: 5548521
               tx-errs: 0
               tx-drop: 0
               tx-fifo: 0
              tx-colls: 0
            tx-carrier: 0
         tx-compressed: 0
```

Properties

Field	Description
name	The name of the Ethernet interface.
mac-address	The MAC address assigned to the Ethernet interface.
op-state	The operational state of the interface, either up or down.
rcv-bytes tx-bytes	The number of bytes transmitted or received without error on the interface.
rcv-packets tx-packets	The number of packets transmitted or received without error on the interface.
rcv-errs tx-errs	The number of errored packets transmitted or received on the interface.

Status provider show commands

Field	Description
rcv-drop tx-drop	The number of packets dropped during the transmit or receive process because the queue was full.
rcv-fifo tx-fifo	The number of times a FIFO underrun occurred during transmit or receive.
rcv-frame	The number of receive packets with bad framing bytes.
rcv-compressed tx-compressed	Total number of compressed packets sent or received by the interface.
rcv-multicast	The number of multicast packets received on the interface.
tx-colls	The number of transmit collisions detected on the interface.
tx-carrier	The number of times the carrier sense was lost during transmit.

show interface-throughput

Purpose

Displays, for each Ethernet interface, the throughput across the interface measured in packets per second (pps) and kilobits per second (kbps), for both transmit and receive. (Each interface has four entries.)

Sample Output

```
NNOS-E> show interface-throughput
name  value   10 second  1 minute   10 minute  1 hour     maximum
----  -----   ---------  --------   ---------  ------     -------
eth0  rx-kbps 0          0          0          0          16
eth0  tx-kbps 10         10         10         10         31
eth0  rx-pps  0          0          0          0          15
eth0  tx-pps  20         20         20         20         28
eth1  rx-kbps 779        388        249        298        22376
eth1  tx-kbps 12309      3924       1004       316        13663
eth1  rx-pps  1163       436        180        134        4051
eth1  tx-pps  1627       574        206        126        1797
eth2  rx-kbps 0          0          0          0          0
eth2  tx-kbps 0          0          0          0          0
eth2  rx-pps  0          0          0          0          0
```

```
eth2  tx-pps  0          0          0          0          1
eth3  rx-kbps 0          0          0          0          0
eth3  tx-kbps 0          0          0          0          0
eth3  rx-pps  0          0          0          0          0
eth3  tx-pps  0          0          0          0          0
```

Properties

Field	Description
name	The name of the Ethernet interface.
value	The measurement for the corresponding interface, either receive (rx) or transmit (tx) packet and speed rates.
10 second	The average rate in the last ten seconds.
1 minute	The average rate in the last one minute.
10 minutes	The average rate in the last ten minutes.
1 hour	The average rate in the last one hour.
maximum	The maximum value recorded since last system boot.

show interfaces

Purpose

Displays the name, IP address, MAC address, and the current operational state of each displays administratively enabled IP interfaces on the system. Interfaces are configured using the interface object.

Sample Output

```
NNOS-E> show interfaces
interface    name          ip-address         op-state      type
---------    ----          ----------         --------      ----
eth0         heartbeat      192.168.0.1/30     up            public
eth1:1       public         215.2.3.0/24       down          public
vx111        Management     172.100.0.10/24    up            public
vx112        Public         10.10.10.10/24     up            public
```

Properties

Field	Description
interface	The interface on which the system is reporting status, either an Ethernet (ethX) or virtual (vx) interface. The name displayed is the unique OS interface name.
name	The name given to the IP interface when it was created (the IP configuration object name).
ip-address	The IP address assigned to the IP interface.
op-state	The operational state of the interface.
type	The type of interface, either public or private. A standard interface is public. IP interfaces configured for virtual firewalls are private.

show ip-counters

Purpose

Displays IP statistics for all interfaces configured for IP routing.

Sample Output

```
NNOS-E> show ip-counters
       forwarding: 1
      default-ttl: 64
      in-receives: 5973190
    in-hdr-errors: 0
   in-addr-errors: 0
   forw-datagrams: 0
in-unknown-protos: 0
      in-discards: 0
      in-delivers: 5636163
     out-requests: 8823298
     out-discards: 0
    out-no-routes: 0
    reasm-timeout: 0
      reasm-reqds: 0
        reasm-oks: 0
      reasm-fails: 0
         frag-oks: 0
```

```
               frag-fails: 0
             frag-creates: 0
```

Properties

Field	Description
forwarding	The state of the interface, either forwarding (1) or not forwarding (2).
default-ttl	The default time-to-live value inserted into IP headers.
in-receives	The number of incoming datagrams received, including those received with errors.
in-hdr-errors	The number of incoming datagrams discarded due to errors in the IP header (e.g., bad checksum, ttl expire, version mismatch, etc.).
in-addr-errors	The number of incoming datagrams discarded because the IP address in the destination field was in error.
forw-datagrams	The number of datagrams forwarded.
in-unknown-protos	The number of datagrams discarded due to an unknown or unsupported protocol.
in-discards	The number of non-errored incoming datagrams that were discarded (e.g., due to congestion).
in-delivers	The number of ICMP, UDP, BOOTP, and TCP incoming datagrams that were successfully delivered to higher-layer protocols.
out-requests	The number of datagrams that local IP protocols requested be transmitted transmission.
out-discards	The number of outgoing error-free datagrams discarded, for example due to buffer overload.
out-no-routes	The number of outgoing datagrams discarded because a valid route could not be found.
reasm-timeout	The maximum number of seconds that the pieces of a fragmented datagram can be held awaiting reassembly before they are discarded.
reasm-reqds	The number of fragments received that required reassembly.

Field	Description
reasm-oks	The number of datagrams successfully reassembled.
reasm-fails	The number of datagrams that could not be reassembled.
frag-oks	The number of datagrams successfully fragmented.
frag-fails	The number of datagrams that were discarded because they could not be fragmented (e.g., because the DoNot Frag flag was set).
frag-creates	The number of datagram fragments created.

show kernel-rule

Purpose

Displays the rules, derived from dos-policies, that reside in the kernel and how AA-SBC acts on them (used primarily for debugging). The output could potentially contain several thousand rules. For simple debugging, use the command to verify that the hit count (Pass) is increasing and that the Drop count is not. An increase in **Forced** is not problematic however. **Forced** indicates the result of rules being set to drop specific types of packets. For example, a **Forced** drop could indicate that a rule was set to drop DTMF packets. In another example, the kernel could drop a packet from further processing until the criteria are met because RTCP has not seen the minimum number of consecutive packets specified in the rule.

Sample Output

```
NNOS-E> show kernel-rule

source           dest                Prot intf        info
------           ----                ---- ----        ----
0.0.0.0:0        255.255.255.255:67  udp  eth2        Pass: 0  Drop: 0
                                                      End
0.0.0.0:0       224.0.0.18:0        vrrp eth2        Pass: 612673  Drop: 0
                                                      End
0.0.0.0:0        172.26.0.56:0       all              Pass: 12954  Drop: 0
                                                      End
```

```
0.0.0.0:0          192.168.217.1:0      all                 Pass: 48321  Drop: 0
                                                            End
0.0.0.0:0          192.168.215.101:0    all                 Pass: 0  Drop: 0
                                                            End
0.0.0.0:0          172.26.0.56:0        udp  eth0           Pass: 50979  Drop: 0
                                                            PortTracker
                                                            End
0.0.0.0:0          192.168.215.101:0    udp  vx0            Pass: 0  Drop: 0
                                                            PortTracker
                                                            End
```

Properties

Field	Description
source	The source IP address of packets dropped. An address of 0.0.0.0 indicates that the system will apply the rule to any IP address. The suffix :0 indicates that the system will apply the rule to any port (TCP or UDP).
destination	The destination IP address of packets dropped.
protocol	The protocol that the incoming traffic is using. Any protocol is valid.
interface	The specific physical interface.
information	A description of how the rule was applied to each source and destination in terms of hits and drops.

show kernel-sip-rules

Purpose

Displays, for all traffic except TLS, filtering at the kernel level instead of at the application layer (use show dos-rules for the application layer). If a DOS attack is in progress, these are the rules that block the attack. The application layer detects the attack and creates the rule; whereas, at the kernel level the attack is actually blocked. The kernel level does not perform decryption. Since TLS is encrypted, handling of the DOS attack is elevated to the **dos-rules** level, where decryption can occur and where the attack is blocked only if all criteria for the block match.

Sample Output

```
NNOS-E> show kernel-sip-rules
rule: 1
packet-count: 624872
remote: any
local: 10.1.208.140
protocol: any
action: filter
match: method: “invite” request-uri: “”
```

Properties

Field	Description
rule	The internal number assigned to the rule in the index of rules.
packet-count	The number of packets received that encountered this rule in this attack.
remote	The remote IP address that is the source of the packet, as specified by the DOS rule. This may be a specific address or “any” to indicate that any sending endpoint that matches the other criteria qualifies for filtering.
local	The destination address of the incoming packets.
protocol	The protocol specified in the policy. If not specified, the value is any.
action	The action taken by the system on matching packets, either: • Filter these SIP packets (throw away). • Alert (allow these SIP packets to go through, but send a log message and SNMP traps).
match	A description of the packet type and fields that are being examined. The method field indicates the message type. If the DOS rule specified header and match strings, the output displays those as well.

show kernel-version

Purpose

Displays detailed information about the kernel currently running.

Sample Output

```
NNOS-E> show kernel-version

version       build        branch        time                         computer
-------       -----        ------        ----                         --------
2.6.11-4-cov  25500:25501S kernel-2.6.11-4 15:42:49 Tue 2007-04-03  dubuc
```

Properties

Field	Description
version	The version number of the kernel currently running.
build	The build number of the kernel currently running.
branch	The branch of the kernel currently running.
time	The time when the kernel was built.
computer	The name of the system on which the kernel was built.

show license-details

Purpose

Displays configurable features based on the license allowances. Features can be managed using the features object, but ultimately are determined by your purchased licensed. Output indicates the objects (class) and property that the license controls. The sample output shows only a piece of representational command output.

Sample Output

```
NNOS-E> show license-details
```

```
name                   class                   property
----                   -----                   --------
INTERNAL BULK LICENSE    accounting
INTERNAL BULK LICENSE    box                    number
INTERNAL BULK LICENSE    cluster                vrrp
INTERNAL BULK LICENSE    database
INTERNAL BULK LICENSE    enterprise             directories
INTERNAL BULK LICENSE    entry                  authentication
INTERNAL BULK LICENSE    entry                  media-type
INTERNAL BULK LICENSE    features
   audio-recording-entities
INTERNAL BULK LICENSE    features
   audio-recording-sessions
INTERNAL BULK LICENSE    features               cpus
INTERNAL BULK LICENSE    features               crypto
INTERNAL BULK LICENSE    features
   dos-protection-entities
INTERNAL BULK LICENSE    log-alert              message-logging
INTERNAL BULK LICENSE    mcafee
INTERNAL BULK LICENSE    mcafee                 update-url
INTERNAL BULK LICENSE    media
INTERNAL BULK LICENSE    media                  anchor
INTERNAL BULK LICENSE    media                  nat-traversal
INTERNAL BULK LICENSE    media                  packet-marking
INTERNAL BULK LICENSE    media                  recording-policy
INTERNAL BULK LICENSE    media                  verify
INTERNAL BULK LICENSE    permissions            debug
INTERNAL BULK LICENSE    policies               dos-policies
INTERNAL BULK LICENSE    preferences            dos-queries
INTERNAL BULK LICENSE    presence               presence-translation
INTERNAL BULK LICENSE    session-config         authentication
INTERNAL BULK LICENSE    session-config         csta-settings
INTERNAL BULK LICENSE    session-config         file-transfer
INTERNAL BULK LICENSE    session-config         forking-settings
INTERNAL BULK LICENSE    session-config         header-settings
INTERNAL BULK LICENSE    session-config         media-type
INTERNAL BULK LICENSE    vsp
INTERNAL BULK LICENSE    vsp
   call-admission-control
INTERNAL BULK LICENSE    vsp                        max-calls-in-setup
```

Properties

Field	Description
name	The name of the license that controls the class and property.

Field	Description
class	The configuration object that the license controls.
property	The property of the object (class) that the license controls.

show licenses

Purpose

Displays summary information for the active license.

Sample Output

```
NNOS-E> show licenses
       name: INTERNAL BULK LICENSE
description: INTERNAL BULK LICENSE
        key: 84420f9a-da13-4107-8833-d00b7d4d751d
    expires:
       file: 84420f9a-da13-4107-8833-d00b7d4d751d.xml
```

Properties

Field	Description
name	The name of the license, as provided.
description	A text field, provided, to help identify the contents of the license.
key	The private key provided to you by Avaya. This key is used for authentication when you contact the licensing server. You need to supply this value to retrieve a modified license, for example, for an extension on the expiration date.

Field	Description
expires	The date at which the license expires. AA-SBC generates an event when it nears that expiration date. You can renew your license by re-executing the license fetch command. The license server verifies that there is a valid license renewal associated with your system ID, and then resets the license expiration to a new date.
file	The name of the actual file on the system that contains the license. The name will be the same as the key in most circumstances.

show location-bindings

Purpose

Displays registration status and location information for each binding of each AOR in the location cache. This command provides information on how to contact an endpoint. The cache is the location information for the local box. Use the show location-bindings command to see bindings for all entries shared throughout the cluster.

Sample Output

```
NNOS-E> show location-bindings
AOR                           STATE        HOST            PORT    TPT
   EXP
---                           -----        ----            ----    ---
   ---
sip:2125551111@vfn.com        registered   172.30.0.208    2057    UDP
   225
sip:2125552222@vfn.com        registered   172.30.0.208    2057    UDP
   220
```

Properties

Field	Description
AOR	The entry in the location cache. Typically, this is the URI associated with the SIP user.
STATE	The state of the AOR.
HOST	The location AA-SBC should use to reach this AOR. Displays as a host name or IP address.
PORT	The contact port on AA-SBC used to reach this AOR.
TPT	The protocol used with this AOR. Note that if the AOR display name begins with “sips,” the protocol must be TLS.
EXP	The number of seconds, as reported in the 200 OK, until the binding expires if it is not renewed.

A binding in the location cache can be in one of 13 states:

State	Description
`requested`	REGISTER request received.
`trying`	REGISTER forwarded and waiting.
`responded`	REGISTER response received.
`aborted`	REGISTER aborted from trying.
`waiting`	Waiting on server busy and will re-register in brief interval.
`challenged`	SIP 401/407 “Auth Required” response has been sent to the endpoint.
`unauthenticated`	Client did not responded to challenge in challenge-timeout period.
`declined`	REGISTER declined with proper code; AA-SBC continues to process subsequent REGISTERs.
`rejected`	All REGISTERs for this binding were rejected with proper code before session was created.

Status provider show commands

State	Description
discarded	All REGISTERs for this binding were discarded silently before session was created.
registered	This binding is valid and registered.
aged	This binding is aged but not deleted.
disconnected	The TCP/TLS connection for this binding is broken.

show location-cache

Purpose

Displays the location database known to the local box. The location cache is the local listing of AORs. Use show location-database to view the shared location service across a cluster.

The output displays state and registry information for each static and learned address of record in the local AA-SBC database of AORs. All location record types are stored in the location cache, a binary tree-based table that contains all location bindings.

This command displays only the AORs; see show location-bindings to display each binding that is associated with the AOR.

Sample Output

```
NNOS-E> show location-cache
AOR                               BOX   STATE        BD   SERVER     H
   ITS   CL
---                               ---   -----        --   ------     -
   ---   --
sip:15554443333@test.babytel.ca   1     registered   2    btel-jim   1
   09    2
sip:15554442222@test.babytel.ca   1     registered   4    btel-jim   1
   41    4
sip:2125551111@voip2.cov.com      1     registered   2    bsoft      7
   4     1
sip:2125552222@voip2.cov.com      1     in-service   2    bsoft      2
   0     0
sip:2125553333@voip2.cov.com      1     registered   1    bsoft      1
   393   0
```

```
sip:2125554444@voip2.cov.com      1     registered   1    bsoft      2
   673   0
sip:jdoe@lcs.companyXYZ.com       1     unregistered 1    Eclipse    2
   0     0
```

Properties

Field	Description
AOR	The entry in the location cache. Typically, this is the URI associated with the SIP user.
BOX	The number of the box that the AOR was registered on (learned from). The output displays "1" for the local box, either standalone or cluster.
STATE	The actual current state of the AOR, including any intermediate states (e.g., WAITING, TRYING). Contrast this to the show location-database command, where the system only displays the final state (e.g. REGISTERD, OUT-OF-SERVICE).
BD	The number of bindings associated with the AOR.
SERVER	The server that the AOR is registered to. If the AOR is registered to the local box, the output displays "Eclipse." Otherwise, it displays the name of the enterprise server that handled the AOR.
HITS	The number of times the AOR was accessed to forward a call, via the dial- or registration-plan (lookups on the AOR).
CL	The number of calls this AOR has participated in—either originated or received.

show location-database

Purpose

Displays the location database, the shared location service across a cluster. This database is used to maintain synchronization of boxes.

The database stores (and therefore this command shows) all learned location bindings; static records are not maintained in the location database as they are managed by configurations.

Sample Output

```
NNOS-E> show location-database
AOR                          BOX    STATE         BD    SERVER    HITS
     CL
---                          ---    -----         --    ------    ----
     --
sip:2125551111@vfn.com        1     registered   0     6         0
sip:2125552222@vfn.com        1     registered   0     6         0
```

Properties

Field	Description
AOR	The learned entry in the location database. Typically, this is the URI associated with the SIP user.
BOX	The number of the box that the learned AOR was registered on (learned from). The output displays "1" for the local box, either standalone or cluster.
STATE	The state of the AOR in the database. This command only displays the final state for the entry (e.g., REGISTERD, OUT-OF-SERVICE), not any intermediate states.
BD	The number of bindings associated with the AOR.
SERVER	The server that the AOR is registered to. If the AOR is registered to the local box, the output displays "Eclipse." Otherwise, it displays the name of the enterprise server that handled the AOR.
HITS	The number of times the AOR was accessed to forward a call, via the dial- or registration-plan (lookups on the AOR).
CL	The number of calls this AOR has participated in—either originated or received.

show location-database-bindings

Purpose

Displays registration status and location information for each binding of each AOR in the location database. This command provides information on how to contact an endpoint. The database stores location information for all boxes across a cluster. Use the show location-bindings command to see bindings for the local box.

Sample Output

```
NNOS-E> show location-database-bindings
AOR                               STATE      HOST             PORT   TPT EXP
---                               -----      ----             ----   --- ---
sip:2125551111@vfn.com            requested  0.0.0.0          2057   UDP 60
sip:2125552222@vfn.com            requested  0.0.0.0          2057   UDP 60
```

Properties

Field	Description
AOR	The entry in the location database. Typically, this is the URI associated with the SIP user.
STATE	The state of the AOR.
HOST	The location AA-SBC should use to reach this AOR. Displays as a host name or IP address.
PORT	The contact port on AA-SBC used to reach this AOR.
TPT	The protocol used with this AOR. Note that if the AOR display name begins with “sips,” the protocol must be TLS.
EXP	The number of seconds, as reported in the 200 OK, until the binding expires if it is not renewed.

show log-targets

Purpose

Displays logging statistics (messages, bytes, and errors) for each configured log target. Use the services object to enable and disable logging for each service.

Sample Output

```
NNOS-E> show log-targets

name                          messages   bytes                errors
----                          --------   -----                ------
file kernel                   60         5913                 0
file messages                 72959      9853391              0
local-database                72982      6570117              0
syslog 192.168.215.1          73         8213                 0
```

Properties

Field	Description
name	The name of the log target.
messages	The total number of messages logged.
bytes	The total number of bytes logged.
errors	The number of errors that were reported for that log target.

show login-sessions

Purpose

Displays all active login session, the type of connection, and the associated user name and permissions.

Sample Output

```
NNOS-E> show login-sessions

started                  type       username            permissions
-------                  ----       --------            -----------
08:18:49 Wed 2006-12-20  console    guest               guest
08:21:46 Wed 2006-12-20  ssh        guest               guest
11:07:14 Wed 2006-12-20  web        guest               guest
```

Properties

Field	Description
started	The time that the login session was initiated.
type	The type of connection to AA-SBC, either: • console—serial console client • ssh—SSH client • telnet—Telnet client • web—AA-SBC Management client • web-service—web services client • monitor—monitor console client
username	The name of the locally configured user logged in to the system. This user was created with the users object.
permissions	The named permissions profile associated with the user. This profile was created with the permissions object.

show master-services

Purpose

Displays each master service and its current configuration. The output includes status of each service regarding mastership and any associated hosts. Master services are configured using the master-services object

Sample Output

```
NNOS-E> show master-services
name           hosted position waiting group host            host-position
----           ------ -------- ------- ----- ----            -------------
accounting     false  2        false   0     192.168.0.2     1
authentication false  2        false   0     192.168.0.2     1
call-failover  false  0        false   0     0.0.0.0         0
cluster-master true   2        false   2     0.0.0.0         2
database       false  2        false   0     192.168.0.2     1
directory      false  2        false   0     192.168.0.2     1
registration   false  2        false   0     192.168.0.2     1
server-load    false  2        false   0     192.168.0.2     1
```

Properties

Field	Description
name	The name of the master service.
hosted	Whether the service is hosted on this box. True indicates that the service is currently hosted on this box, false that it is not hosted on this box (or not configured). The **host-box** property within each master-services object defines the primary box for that service. If you configured backup boxes, the master service would be hosted on the backup in the event of primary failure.

Field	Description
position	The position this box is in for the hosting responsibility of the master service. A value of 0 indicates that the box is not in the list of eligible boxes (or the service is not configured). If the position for a hosted service is not in position 1, it indicates that the service has failed over to a backup box. The positions of the boxes can be displayed at the command line by typing **show -v** from within the master-service object. The number in brackets next to the **host-box** property lists the position of the device in the configuration.
waiting	The state of any host take over process. A value of true indicates that the listed host is currently attempting to take over the service.
group	The VRRP group that the master service is associated with.
host	The IP address of the box currently hosting the service. A value of 0.0.0.0 indicates the local box.
host-position	The position of the host box in the list of eligible boxes. See the description of the position field for more information.

show media-ports-held

Purpose

Displays any media ports being held by AA-SBC. A port is held if AA-SBC suspects the port or knows it to be bad. The system clears the port status as appropriate.

Sample Output

```
NNOS-E> show media-ports-held
     port: 21448
remaining: 0 days 00:00:15
```

Properties

Field	Description
port	The number of the port.
remaining	The amount of time remaining until the system stops holding the port.

show media-ports-summary

Purpose

Displays a summary of media port use and configuration for each IP interface on which media ports are configured. The base-port and count are set with the IP interface media-ports object. You set the per-processor limits on the number of ports (and thus, the number of active calls) available for media anchoring at any one time using the object media-anchor-limits **port-limits** properties. The output also displays the port status and availability.

Sample Output

```
NNOS-E> show media-ports-summary
ip-address      base-port count busy  held  free
----------      --------- ----- ----  ----  ----
10.1.34.160     20000     5000  0     0     5000
127.0.0.1       10000     55000 8     0     54992
172.26.0.155    20000     5000  4     0     4996
```

Properties

Field	Description
ip-address	The configured IP address for the interface.
base-port	The base or starting port number to use for the port pool on the corresponding interface.
count	The total number of ports available for the media port pool on the corresponding interface.

Field	Description
busy	The number of ports in use within the IP interface port pool.
held	The number of ports being held within the IP interface port pool. A port is held if AA-SBC suspects the port or knows it to be bad. The system clears the port status as appropriate.
free	The number of ports available within the IP interface port pool.

show media-scanner-interval

Purpose

Displays media scanner intervals. The media-scanner monitors the signal strength and duration of the received audio to divide it into intervals.

Sample Output

```
NNOS-E> show media-scanner-interval

session-id        streamIndex  call-leg   start-time   duration  level
   flags
--------------------------------------------------------------
0x4c2d14240838d8f    0           1          9         1600      -54
   short-pause
0x4c2d14240838d8f    0           1          1629      200       -18
   short-talk
0x4c2d14240838d8f    0           1          1829      100       -40
   short-pause
0x4c2d14240838d8f    0           1          1929      200       -20
   short-talk
0x4c2d14240838d8f    0           1          2129      100       -46
   short-pause
0x4c2d14240838d8f    0           1          2229      600       -26
   short-talk
0x4c2d14240838d8f    0           1          2829      2000      -44
   long-pause
```

```
0x4c2d14249b10e50    0            1            12          2000        -52
    long-pause
0x4c2d1424de7598e    0            1            12          1400        -52
    short-pause
0x4c2d1424de7598e    0            1            1432        500         -32
    short-talk
0x4c2d1424de7598e    0            1            1932        100         -44
    short-pause
0x4c2d1424de7598e    0            1            2032        200         -24
    short-talk
```

Properties

Field	Description
session-id	The unique ID of the media scanner session.
streamIndex	An index number indicating a particular stream of media for the session. For example, a normal phone call has one audio stream. So stream is one. And a video call has one audio stream (stream=1) and one video stream (stream=2).
call-leg	An index for a call leg indicating the direction of the call.
total-duration	The duration of the call.
current-flags	The current interval a call is in. The following are possible interval values: short-pause—small gaps between spoken words short-talk—short talk spurts long-talk—long, uninterrupted speech long-pause—signal to the NNOS-E that the media scanning is complete stable-tone—signal to the NNOS-E that the media scanning is complete
total-flags	The total number of intervals that have occurred.

show media-scanner-summary

Purpose

Displays media scanner settings. The media-scanner monitors the signal strength and duration of the received audio to divide it into intervals.

Sample Output

```
NNOS-E> show media-scanner-summary
session-id      streamIndex  call-leg  total-duration  current-flags
    total-flags
--------------------------------------------------------------
```

```
0x4c2d14240838d8f    0           1          4800          long-pause
   short-pause+short-tal
0x4c2d14249b10e50    0           1          2000          long-paus
x4c2d1424de7598e    0           1          5800          long-pause
   short-pause+short-talk
```

Properties

Field	Description
session-id	The unique ID of the media scanner session.
streamIndex	An index number indicating a particular stream of media for the session. For example, a normal phone call has one audio stream. So stream is one. And a video call has one audio stream (stream=1) and one video stream (stream=2).
call-leg	An index for a call leg indicating the direction of the call.
start-time	The time when the call started.
duration	The duration of the call.
level	The measured dbM level of the sound we received during a given time interval.
flags	The intervals that have occurred.

show media-stream-addresses

Purpose

Displays the session ID, IP address, and port for each leg of a call in a media stream. The output also displays details on each segment of the call leg.

The rows of the display are ordered from top to bottom, approximating the stages a media packets flows through. For basic calls, the rows marked with call-leg=1 describe the flow of media packets from the calling phone towards the answering phone (the forward path). The rows marked with call-leg=2 describe the flow of media from the answering phone towards the calling phone (the reverse path).

The following are some troubleshooting tips:

- Command output displaying all zeroes in the **address** fields:

  Check that media anchoring is not disabled for the session. Verify that all media objects that apply (under policy, dial-plan, and default-session-config) have the anchoring property set to **enable**.

- One or more **address** entries for **type peer-source** displays zero:

  This indicates that AA-SBC did not receive RTP packets on the problematic call-leg. Routing or other network issues may be preventing the phone from reaching AA-SBC. Or, if the phone is behind a NAT, verify that the **symmetricRTP** property, in the nat-traversal object, is set to **true**.

- Output appears fine

  If neither **peer-source** lines are zero, AA-SBC is anchoring media in both directions. In this case, the packets sent by AA-SBC towards one or both of the phones are not being received by the respective phones. Routing or other network issues may be the problem.

Sample Output

```
NNOS-E> show media-stream-addresses
session-id          stream  call-leg  type           origin      address
----------          ------  --------  ----           ------      -------
0x4c2702a100005c6   1       1         peer-source
   rtp          172.30.0.18:7350
                                      anchor-dest

   media-port   172.26.0.15:22264
```

```
                                         anchor-source media-port  172.26.
            0.15:22194
                                         peer-dest     sdp         172.30.
            0.20:37520
                          2              peer-source   rtp         172.30.
            0.20:37520
                                         anchor-dest   media-port  172.26.
            0.15:22194
                                         anchor-source media-port  172.26.
            0.15:22264
                                         peer-dest     sdp         172.30.
            0.18:7350
```

Properties

Field	Description
session-id	The unique internal identifier assigned to the session.
stream	The index number of the media-stream. Voice calls have just one stream, with index equal to 1. Video calls have two streams—the audio stream is typically index equal to 1 and the video stream is typically index equal to 2.
call-leg	The index number of the call leg. For basic calls call-leg 1 describes the media flowing in the direction from the calling party (the SIP From: URI) towards the answering party (the SIP To: URI). Call-leg 2 describes the reverse direction—from the answering party towards the calling party.

Field	Description
type	The role the address plays in the call. The **type** column can have the following values: • **peer-source**—the address of an RTP peer (either calling or answering party) that is the source of RTP packets received by the system. This type of address is always learned from the IP and UDP headers of received RTP packets and has an **origin** value (see below) of **rtp**. • **peer-dest**—the address of an RTP peer (either calling or answering party) that is the destination of RTP packets sent by the system. This type of address is learned either from SDP (**origin** value of **sdp**) or from symmetric RTP (**origin** value of **symmetric-rtp**). • **anchor-dest**—the system address that receives packets from an RTP peer. This type of address has an **origin** value of **media-ports** or **near-end-nat**. • anchor-source—the system address that sends packets to an RTP peer. This type of address has an **origin** value of **media-ports**.
origin	The protocol source of the address (how the address was determined). The **origin** column can have the following values: • **sdp** (Session Description Protocol)—these addresses are learned from the “c=” lines in received SDP message bodies. • **media-port**—these addresses are allocated from media-ports configured under the IP interface. • **rtp** (Real Time Protocol)—these addresses are learned from the IP/UDP headers of received RTP packets. • **symmetric-rtp**—these addresses are learned from the IP/UDP headers of received RTP packets from the reverse media direction. • **near-end-nat**—these addresses are determined by the near-side-nat configuration under the IP interface.
address	The IP address and UDP port related to media anchoring on the session.

show media-stream-client-sessions

Purpose

Displays each signaling session that has media resources allocated on a media-proxy (a media stream server). The output is primarily used for debugging and troubleshooting.

Sample Output

```
NNOS-E> show media-stream-client-sessions

client-session-id server-id server-session-id
----------------- --------- -----------------
0x4c105504fc789e0 0.0.0.0   0x4c105504fc94123
```

Properties

Field	Description
client-session-id	The unique internal ID of the media stream client session.
server-id	The IP address of the media stream server.
server-session-id	The ID of the media stream server session.

show media-stream-counts

Purpose

Displays how many voice sessions are up on AA-SBC at a given time.

Sample Output

```
NNOS-E> show media-stream-counts

client-id   server-id   sessions
---------   ---------   --------
0.0.0.0     0.0.0.0     2
```

Properties

Field	Description
client-id	The IP address of the client.
server-id	The IP address of the media stream server.
sessions	The number of sessions involved in the call. A call can be made up of multiple sessions. For example, in a B2B configuration, the system receives the call, terminates the session, and then creates a new session and sends it on to the recipient. In this case, there are multiple sessions associated with the call.

show media-stream-server-sessions

Purpose

Displays each media stream server session created for a signaling session on a signaling node (a media stream client). The output is primarily used for debugging and troubleshooting.

Sample Output

```
NNOS-E> show media-stream-server-sessions

server-session-id client-id client-session-id
----------------- --------- -----------------
0x4c1055063573eb0 0.0.0.0   0x4c105506355a06d
```

Properties

Field	Description
server-session-id	The unique internal ID of the media stream server session.
client-id	The IP address of the media stream client (the signaling node).
client-session-id	The ID of the media stream client session.

show media-stream-stats

Purpose

Displays a count of transmit and receive packets for each call leg in a media stream.

Sample Output

```
NNOS-E> show media-stream-stats

session-id         stream call-leg address
                  receive-packets transmit-packets
----------         ------ -------- -------
                  --------------- ----------------
0x4c105506355a06d 1      1        192.168.215.103:22566 21407
             21437
                         2        192.168.215.103:22394 21437
             21407
```

Properties

Field	Description
session-id	The ID of the media stream session.
stream	An index number indicating a particular stream of media for the session. For example, a normal phone call has one audio stream. So stream is one. And a video call has one audio stream (stream=1) and one video stream (stream=2).
call-leg	An index for a call leg indicating the direction of the call.
address	The IP address and media port of the media stream server where it receives and transmits packets.
receive-packets	The number of packets the media stream server has received.
transmit-packets	The number of packets the media stream server has transmitted.

show memory-failures

Purpose

Indicates whether any memory allocation failures have occurred on the box. If the output indicates a failure, you can troubleshoot possible reasons the box is out of memory (e.g., a configuration problem, a memory leak, etc.).

Sample Output

```
NNOS-E> show memory-failures
Memory allocation failures:
---------------------------------------------------------------------
Process  Address  Failures  OldestFail  NewestFail  Smallest  Largest
   ------------------------------------------------------------------
   --
   manager  08271b60     1     00:00:03     00:00:03   1048612  10486
   12
   manager  08272834     1     00:00:03     00:00:03     65536    655
   36
   manager  08272d0b     1     00:00:03     00:00:03        32
   32
   manager  08272efb     1     00:00:03     00:00:03        32
   32
   manager  08277353     1     00:00:03     00:00:03      1024     10
   24
   manager  082774dc     1     00:00:03     00:00:03         1
    1
   manager  082760fb     1     00:00:03     00:00:03       256      2
   56
   manager  08276b1f     1     00:00:03     00:00:03       699      6
   99
   manager  08276b9f     1     00:00:03     00:00:03       699      6
   99
   ------------------------------------------------------------------
   --
```

Properties

Field	Description
Process	The system process that generated the memory failure.
Address	The address of the failure.

Field	Description
Failures	The number of failures at that address.
Oldest Fail	The amount of time that has passed since the first failure at this address occurred. Use the verbose form of the command to see a timestamp for the failure.
Newest Fail	The amount of time that has passed since the most recent failure at this address occurred. Use the verbose form of the command to see a timestamp for the failure.
Smallest	The smallest memory allocation size that failed at that address.
Largest	The largest memory allocation size that failed at that address.

show netfilter

Purpose

Displays the state of the Linux firewall, which is effected when a service (e.g., SSH, Telnet) is configured. The output displays entries through the firewall that AA-SBC created to allow service traffic to be received. Netfilter rules are automatically generated by the services as they are configured on an interface.

Sample Output

```
NNOS-E> show netfilter

index  packets  policy  intf   proto source     destination        mat
   ch
-----  -------  ------  ----   ----- ------     -----------        ---
   --
1      244161   permit         tcp   0.0.0.0:0  0.0.0.0:0          sta
   te:EST,REL
2      20846    permit         udp   0.0.0.0:0  0.0.0.0:0          sta
   te:EST,REL
3      437      permit         all   0.0.0.0:0  127.0.0.1:0
4      0        permit  eth0   tcp   0.0.0.0:0  172.26.0.153:23    fla
   gs:SYN state:NEW
5      1        permit  eth0   tcp   0.0.0.0:0  172.26.0.153:22    fla
   gs:SYN state:NEW
```

```
6       0        permit  eth0   udp   0.0.0.0:0  172.26.0.153:161   sta
    te:NEW
7       0        permit  eth0   udp   0.0.0.0:0  172.26.0.153:123   sta
    te:NEW
8       2077     permit  eth0   udp   0.0.0.0:0  255.255.255.255:67 sta
    te:NEW
--more--
```

Properties

Field	Description
index	The system-generated identifier for the netfilter rule. Each rule has a unique index, and index numbers are assigned in the order in which the services were configured on the system. Rules are processed in the order of the index numbers.
packets	The number of packets effected by the rule.
policy	The action that matching the rule results in, either permit or deny. The final rule entry is always to deny all. (If a packet did not match any other rule that determines its outcome, the system denies the packet.) All rules automatically created by the system in response to a service configuration are given a policy of **permit**.
intf	The interface on which the rule applies.
proto	The protocol used with the service that the rule represents. The protocol is determined by the service configuration.
source	The source address and port of the packet to match against. An entry of 0.0.0.0:0 represents a wildcard (match any).
destination	The destination address and port of the packet to match against. An entry of 0.0.0.0:0 represents a wildcard (match any).
match	The string that describes what the rule is matching against.

show network-settings

Purpose

Displays the current system network settings for TCP properties. These properties are set with the network object.

Sample Output

```
NNOS-E> show network-settings
----------------------------------------------------------------------
Configuration Item        Current Value
----------------------------------------------------------------------
tcp-keepalive-time        600 seconds
tcp-keepalive-probes      5
tcp-keepalive-interval    6 seconds
tcp-max-syn-backlog       1024
tcp-synack-retries        5
tcp-syncookies            enabled
tcp-fin-timeout           60 seconds
----------------------------------------------------------------------
```

Properties

Field	Description
tcp-keepalive-time	The time, in seconds, that an established TCP connection can remain idle before the system sends a keepalive to the client. The idle time expiration initiates the keepalive process.
tcp-keepalive-probes	The number of unanswered TCP keepalive probes that are allowed before the system disconnects an idle session.
tcp-keepalive-interval	The time, in seconds, that the system waits for a response from a keepalive probe before ending the next one. The system continues to send probes until it has sent the number specified in the **tcp-keepalive-probes** property.
tcp-max-syn-backlog	The maximum number of queued (unacknowledged) connection requests allowed before the system begins dropping requests. This value is set to help prevent a TCP SYN flood attack.

Field	Description
tcp-synack-retries	The number of times the system retransmits a SYN-ACK in response to a SYN. If the number of retries is reached without a successful response, the system deletes the new connection from the table.
tcp-syncookies	Specifies whether SYN cookie support in the kernel is enabled or disabled. See the tcp-syncookies property in the network object for more information.
tcp-fin-timeout	The number of seconds the system waits for a final FIN packet before forcibly closing the socket. The system uses the FIN packet to disconnect a TCP connection, whether it's idle or not.

show ntp

Purpose

Displays statistics relating to the Network Timing Protocol and AA-SBC. You can configure AA-SBC as both an ntp-client and an ntp-server.

Sample Output

```
NNOS-E> show ntp
requests: 174
responses: 173
discards: 0
adjustments: 71
last-adjustment: 10:38:27 Wed 2007-04-11
maximum-adjustment: 2180 milliseconds
average-adjustment: 733 milliseconds
server-requests: 411
server-responses: 411
server-discards: 0
```

Properties

Field	Description
requests	The number of NTP requests the system made when acting as an NTP client.
responses	The number of responses the system NTP client received.
discards	The number of responses the system NTP client received but did not accept.
adjustments	The number of changes the system made to its internal time based on an NTP response.
last-adjustment	The time and date of the last adjustment.
maximum-adjustment	The largest adjustment to the box time the system made based on an NTP response.
average-adjustment	The average adjustment time to the box the system made based on all NTP responses.
server-requests	The number of NTP requests the system received when acting as an NTP server.
server-responses	The number of responses the system NTP server sent out.
server-discards	The number of requests that the system NTP server discarded.

show policies

Purpose

Displays, for each active policy, the number of rules it contains and the number of AA-SBC elements that use that policy. You can configure policy from a variety of places. See Session configuration objects for more information.

Sample Output

```
NNOS-E> show policies

           name: default
number-of-rules: 10
```

```
          inclusions: 0

                name: to
     number-of-rules: 0
          inclusions: 0

                name: from
     number-of-rules: 0
          inclusions: 0
```

Properties

Field	Description
name	The name of the policy.
number-of-rules	The number of rules that comprise the policy.
inclusions	The number of times the policy was triggered (from the piece of the configuration in which it is referenced). For example, you could reference a policy in the vsp\enterprise\unknown-server-policy object. This value would indicate the number of times there was a user in an incoming message that was defined in a configured directory. Otherwise, if there is no such user, the policy is not used and the inclusions indicate zero.

show processes

Purpose

Displays status for each process that is part of AA-SBC operations. If you have debug permissions, you can use the top (NNOS-E>) command to display all processes running on the box.

Sample Output

```
NNOS-E> show processes
process   id        condition  run-level  starts  uptime            fds
-------   --        ---------  ---------  ------  ------            ---
monitor   6538      running    7           1      0 days 01:18:11   26
manager   6686      running    7           1      0 days 01:18:11   58
SIP       6763      running    7           1      0 days 01:18:09   66
media     6764      running    7           1      0 days 01:18:09   26
```

```
auth      0         idle       init        0      0 days 00:00:00  0
reg       6765      running    7           1      0 days 01:18:09  26
dir       0         idle       init        0      0 days 00:00:00  0
web       6767      running    7           1      0 days 01:18:09  13
WS        6768      running    7           1      0 days 01:18:09  15
acct      0         idle       init        0      0 days 00:00:00  0
dos       0         idle       init        0      0 days 00:00:00  0
SSH       6769      running    none        1      0 days 01:18:09  6
```

Properties

Field	Description
process	The name of the system-specific process.
id	An internal ID number, assigned at startup. This ID is analogous to the Linux process ID.
condition	Possible process conditions are: • **idle**—the process did not start or is administratively disabled. Check the master-services object to determine if the process is configured. • **running**—the process is running. • **dead**—the process cannot be restarted. To determine the reason for failure, use the show faults command and check the event logs. • **disabled**—the process will not be started. Check to see whether it is unavailable due to licensing restrictions.
run-level	The process run level, which indicates its state. A process starts off at **init**, and then proceeds from 0 up through 7, at which point it is fully operational. If the process displays **none**, it does not participate in the run level mechanism.
starts	The number of times that the process started (or failed and restarted). For any value other than 0 or 1, use the show faults command and check the event logs for information relating to that process. This counter clears on system restart.
uptime	The length of time the process has been running. The timer restarts when the process boots.
fds	The file descriptor for the process. There is a file descriptor for each open file and socket.

show radius-auth

Purpose

Display configuration information, status, and count and speed statistics for each configured radius-group object. AA-SBC resets all statistics reboot.

Sample Output

```
NNOS-E> show radius-auth

Status for RADIUS group '172.26.0.147' (round-robin):
-------------------------------------------------------------------------------
Server Name        State     | Out  Pending  Requests  Accepts  Rejects  Errors
-------------------------------------------------------------------------------
172.26.0.147       Healthy   |   0        0        19        3        0      16
-------------------------------------------------------------------------------
Totals:                      |   0        0        19

Status for RADIUS group 'Boston' (fail-over):
-------------------------------------------------------------------------------
Server Name        State     | Out  Pending  Requests  Accepts  Rejects  Errors
-------------------------------------------------------------------------------
boston             Idle      |   0        0         0        0        0       0
127.0.0.1          Idle      |   0        0         0        0        0       0
-------------------------------------------------------------------------------
Totals:                      |   0        0         0        0        0       0
```

Properties

Field	Description
Status for RADIUS group...	The name of the group reported on as well as the RADIUS group authentication operational algorithm.
Server Name	The name or IP address that identifies the server that is part of this RADIUS group.

Field	Description
State	The state of the RADIUS server, either: • **Idle**—server is enabled, but has not received traffic. • **Disabled**—the server is disabled in the configuration. • **Healthy**—the server is responding normally to system requests. • **Failing**—the server has not responded to some system requests, but not enough to trigger a fail-over (if configured). • **Failed**—the server has failed to respond to too many requests and the system has determined that it is down. If the RADIUS group is configured with fail-over mode, and a backup server is configured, then the system stops sending requests to this server and begins forwarding requests to the next. • **BadSecret**—there is an error with the shared secret configured for this server is incorrect.
Out	The number of outstanding requests; the requests that the system has sent to the RADIUS server without a response.
Pending	The number of requests that have been generated but have not yet been sent to the server. The server's **window** setting defines the number of allowed requests. Requests generated once that threshold has been reached are counted as pending.
Requests	The total number of authentication requests generated.
Accepts	The number of times the RADIUS server has accepted a request, indicating that the user has the correct password.
Rejects	The number of times the RADIUS server has rejected a request, indicating that either the user has an incorrect password or the shared secret isn't right.
Errors	The number of request errors.

show registration-admission-control

Purpose

Displays settings and statistics for registration admission control on this VSP (REGISTER requests). See the admission-control object for more complete descriptions of all configurable settings.

Sample Output

```
NNOS-E> show registration-admission-control
                                    name: default
          registration-admission-control: enabled
                       max-registrations: 30000
    pending-registrations-high-watermark: 500
     pending-registrations-low-watermark: 10
 pending-registrations-dynamic-threshold: 500
                        cpu-monitor-span: 20 seconds
                    cpu-monitor-interval: 10 seconds
                         average-sip-cpu: 0 %
        registrations-high-cpu-threshold: 90 %
         registrations-low-cpu-threshold: 50 %
                   total-client-bindings: 0
               registrations-in-progress: 0
          registrations-most-in-progress: 0
                  registrations-sessions: 0
             processed-new-registrations: 0
         processed-waiting-registrations: 0
      processed-challenged-registrations: 0
           processed-other-registrations: 0
  suppressed-registrations-this-interval: 0
  suppressed-registrations-last-interval: 0
            suppressed-new-registrations: 0
        suppressed-waiting-registrations: 0
     suppressed-challenged-registrations: 0
             last-register-suppressed-at:
           discarded-other-registrations: 0
              last-register-discarded-at:
            edp-transactions-in-progress: 0
```

Properties

Field	Description
name	The name of the VSP whose status is displayed.
registration-admission-control	The state of registration admission control for this VSP—whether it is enabled or disabled.
max-registrations	The setting for the maximum number of registrations allowed in the location cache. When the system reaches the maximum registration count, registrations are denied until the number falls below this threshold. See the admission-control **max-registrations** property.
pending-registrations-high-watermark	The hard limit set for the number of in-progress registrations allowed before the system suppresses all registrations. See the admission-control **pending-registrations-high-watermark** property.
pending-registrations-low-watermark	Sets a hard limit for the number of in-progress registrations allowed before the system begins registration suppression. See the admission-control **pending-registrations-low-watermark** property
pending-registrations-dynamic-threshold	The current dynamic limit derived for the number of in-progress registrations allowed. The registration dynamic threshold is calculated based on the admission-control **pending-registrations-high-watermark** property.
cpu-monitor-span	The number of seconds over which the system calculates the total system CPU average. See the admission-control **cpu-monitor-span** property.
cpu-monitor-interval	The frequency, in seconds, with which the system calculates the total system CPU average for the last span. See the admission-control **cpu-monitor-interval** property
average-sip-cpu	The current average CPU usage.

Field	Description
registrations-high-cpu-threshold	The upper threshold, as a percentage, for registration processing average CPU usage. See the admission-control **registrations-high-cpu-threshold** property.
registrations-low-cpu-threshold	Sets the low-end threshold, as a percentage, for registration processing average CPU usage. See the admission-control **registrations-low-cpu-threshold** property.
total-client-bindings	The total number of bindings in the location cache.
registrations-in-progress	The number of registrations that the system is currently processing.
registrations-most-in-progress	The highest number of registrations that the system has had in process at any one time.
processed-new-registrations	The number of new registrations that the system successfully processed since the last system boot.
processed-waiting-registrations	The number of registrations for which the system received a REGISTER but the server was busy. As a result, the system responded locally, inserting the registration plan route **min-client-expiration** time.
processed-challenged-registrations	The number of challenge requests that the system processed since the last system boot.
processed-other-registrations	The number of other types of requests that the system processed since the last system boot.
suppressed-registrations-this-interval	The number of REGISTER requests to which the system responded locally instead of delegating in the current interval. The interval is set with the admission-control **cpu-monitor-interval** property
suppressed-registrations-last-interval	The number of REGISTER requests to which the system responded locally instead of delegating in the previous interval. The interval is set with the admission-control **cpu-monitor-interval** property
suppressed-new-registrations	The number of new registrations that the system successfully processed locally (instead of delegating) since the last system boot.

Status provider show commands

Field	Description
suppressed-waiting-registrations	The number of registrations for which the system received a REGISTER but the server was busy. As a result, the system responded locally, inserting the registration plan route **min-client-expiration** time.
last-register-suppressed-at	The time at which the last REGISTER request was suppressed.
discarded-other-registrations	The number of registrations discarded that were not of type new, waiting, or challenge.
last-register-discarded-at	The time at which the last REGISTER request was discarded.
edp-transactions-in-progress	The number of Expiration Discovery Process (EDP) transactions that the system is currently processing. See the admission-control **pending-edp-transaction** properties for information on setting transaction thresholds; see the registration-plan route **edp** property for more information on EDP.

show registration-arbitration

Purpose

Displays a summary of each configured registration-plan arbiter entry, its match criteria and rules (and other configuration elements), and the number of times AA-SBC has applied the plan. The arbiter contains an ordered set of rules that configure different cost-based routing algorithms, which AA-SBC uses to select where to forward REGISTER, SUBSCRIBE, and NOTIFY requests.

Sample Output

```
NNOS-E> show registration-arbitration

plan-name: abc
     type: default
    match: !*
      pri: 100
    rule1:
    rule2:
     hits: 0
```

Properties

Field	Description
plan-name	The name of the arbiter plan entry.
type	The criteria for matching entries in the arbitration table. The matching arbiter is then applied, determining the calculation the system performs. The type field is derived from the **subscriber-match** property setting in the arbiter object.
match	The string to match in the USER and/or HOST fields of the FROM URI in order for the system to apply the plan configuration to requests containing the prefix. The **match** field is derived from the **subscriber-match** property specific string match setting in the arbiter object.
pri	The priority (order of preference) for this registration-plan arbiter entry. This property overrides the default behavior (most specific match) and sets a preference based on the **subscriber-match** property.
rule1	The first rule applied to determine server selection, set with the **rule** property of the arbiter object.
rule2	The second rule applied to determine server selection, set with the **rule** property of the arbiter object.
hits	The number of times this arbiter entry was matched on and applied.

show registration-plan

Purpose

Displays a summary of each *configured* (but not necessarily active) registration-plan entry, its match criteria and peer (and other configuration elements), and the number of times AA-SBC has applied it to forward a call. Use the show registration-plan command to see all *active* registration plans. The output is from the registration routing table, which defines how AA-SBC proxies registrations (handles incoming requests).

Note that the command output includes an entry that is automatically generated by the system. The final entry, a **type** of **domain** and a **match** on the configured VSP domain name is the AA-SBC conversion of the registration service into a registration plan.

Sample Output

```
NNOS-E> show registration-plan
plan-name      type      match           min   pri  peer-name      action   hits
---------      ----      -----           ---   ---  ---------      ------   ----
vfn             default   !*               2   100                 accept   144
vfn.com         domain    vfn.com          7   100  fn.com         accept   12
978             phone     sip:1978.*@.*        50   abc.com        delegate 0
companyXYZ.com  domain    companyXYZ.com  14   100  companyXYZ.com accept   0
```

Properties

Field	Description
plan-name	The name of the route or source-route registration plan entry.
type	The criteria for matching entries in the routing table. The system then applies the appropriate normalization plan to matching entries. The **type** field is derived from either the **to-uri-match** (route) or **source-match** (source-route) property.

Field	Description
match	If contributed through a route object entry, the string to match in the USER and/or HOST fields in the SIP header in order for the system to apply the entry normalization plan to calls containing the prefix. If contributed through a source-route object entry, the match criteria for the source of the SIP message. The system then sets the next-hop server (defined with the **peer** property) for all traffic that matches this configured source. The **match** field is derived from either the **to-uri-match** (route) or **source-match** (source-route) property. If type is condition-list, the match is derived from the priority plus plan name.
min	The minimum number of digits required for a match on a phone prefix, if configured. In some cases, the system calculates a value for other types of matches based on the number of characters (including wild cards). In some cases it displays as-is. The value is only meaningful to a phone-prefix match, however.
pri	The priority (order of preference) for this registration-plan entry. This property overrides the default behavior (most specific match) and sets a preference based on the **subscriber-match** property.
peer-name	A statically entered peer referenced through the **peer** property.

Field	Description
action	The action that the system is configured to take when a match occurs. The following describes each of the possible actions: • **accept**—accepts the registration locally, functioning as a registrar. • **delegate**—forwards the REGISTER to an upstream SIP proxy (provider) and resets the contact to itself. • **forward**—forwards the REGISTER, unchanged, to the server specified in the header. • **redirect**—sends a response to the client with instructions to resend the REGISTER to a different server. • **tunnel**—creates an OC client-to-LCS server tunnel, via the registration plan, that you can then load balance across. • **discard**—silently discards REGISTER requests matching this registration plan or AOR. • **block**—rejects calls matching this registration plan or AOR with either a “603 Declined” message or with configured text.
hits	The number of times this registration-plan entry was matched on and applied.

show registration-routing

Purpose

Displays all active entries in the registration routing table, which defines how AA-SBC handles incoming REGISTER requests. The output displays a summary of each *active* registration-plan entry, its match criteria and peer (and other configuration elements), and the number of times AA-SBC has applied the plan to forward a call. Use the show registration-plan command to see all *configured* registration plans.

Note that the command output includes an entry that is automatically generated by the system. The final entry, a **type** of **domain** and a **match** on the configured VSP domain name is the AA-SBC conversion of the registration service into a registration plan.

Sample Output

```
NNOS-E> show registration-routing
plan-name        type      match            min   pri  peer-name      ac
   tion  hits
---------        ----      -----            ---   ---  ---------      --
   ----  ----
vfn              default   !*               2     100                 ac
   cept  144
vfn.com          domain    vfn.com          7     100  vfn.com        ac
   cept  12
companyXYZ.com   domain    companyXYZ.com   14    100  companyXYZ.com ac
   cept  0
```

Properties

Field	Description
plan-name	The name of the route or source-route registration plan entry.
type	The criteria for matching entries in the routing table. The system then applies the appropriate normalization plan to matching entries. The **type** field is derived from either the **to-uri-match** (route) or **source-match** (source-route) property.

Field	Description
match	If contributed through a route object entry, the string to match in the USER and/or HOST fields in the SIP header in order for the system to apply the normalization plan to calls containing the prefix. If contributed through a source-route object entry, the match criteria for the source of the SIP message. The system then sets the next-hop server (defined with the **peer** property) for all traffic that matches this configured source. The **match** field is derived from either the **to-uri-match** (route) or **source-match** (source-route) property. If type is condition-list, the match is derived from the priority plus plan name.
min	The minimum number of digits required for a match on a phone prefix, if configured. In some cases, the system calculates a value for other types of matches based on the number of characters (including wild cards). In some cases it displays as-is. The value is only meaningful to a phone-prefix match, however.
pri	The priority (order of preference) for this registration-plan entry. This property overrides the default behavior (most specific match) and sets a preference based on the **to-uri-match** (route) or **source-match** (source-route) property.
peer-name	A statically entered peer referenced through the **peer** property of the route or source-route object.

Field	Description
action	The action that the system is configured to take when a match occurs. The following describes each of the possible actions: • **accept**—accepts the registration locally, functioning as a registrar. • **delegate**—forwards the REGISTER to an upstream SIP proxy (provider) and resets the contact to itself. • **forward**—forwards the REGISTER, unchanged, to the server specified in the header. • **redirect**—sends a response to the client with instructions to resend the REGISTER to a different server. • **tunnel**—creates an OC client-to-LCS server tunnel, via the registration plan, that you can then load balance across. • **discard**—silently discards REGISTER requests matching this registration plan or AOR. • **block**—rejects calls matching this registration plan or AOR with either a “603 Declined” message or with configured text.
hits	The number of times this registration-plan entry was matched on and applied.

show registration-service

Purpose

Displays status and statistics for each configured registration service (a registrar that can process REGISTER requests and add AORs updates to the location services database). Enable a domain to act as a registration service using the registration-service object.

Sample Output

```
NNOS-E> show registration-service
domain-name          admin     max-expiration      min-expiration
-----------          -----     --------------      --------------
vfn.com              enabled   90                  30
```

Properties

Field	Description
domain-name	The name of the domain that is enabled to act as the registration service. This is the configured VSP domain name.
admin	The administrative state of the registration service. If disabled in either the **registration-service** or session-config **registration** object, the system rejects any REGISTER request sent to the registration service.
max-expiration	The maximum time (in seconds) to elapse before a client REGISTER request becomes invalid and the registration information is removed from the location cache. A value indicates that the registration-service configuration overwrites the maximum value; **as-requested** indicates that the system maintains the client value.
min-expiration	The minimum time (in seconds) to elapse before a client REGISTER request becomes invalid and the registration information is removed from the location cache. A value indicates that the registration-service configuration overwrites the maximum value; **as-requested** indicates that the system maintains the client value.

show restart-status

Purpose

Displays the state of a controlled install or controlled restart of the system. Use this to check on the progress of these controlled install or restart actions. You can also use this command when you have used the install **file** or **url controlled** action to upgrade boxes in a cluster.

Sample Output

```
NNOS-E> show restart-status

started:
  state: idle
```

Properties

Field	Description
started	The time that the system install or restart was started.
state	The system state with regard to the install or restart action. For example, idle, loading, or restarting. The state may also indicate exactly what is loading or restarting, such as “loading 172.26.0.48” or “restarting sip stack.”

show routing

Purpose

Displays the generic routing table, which contains destination network and host addresses. These addresses are either configured IP interfaces (added through the ip object) or static routes (added with the routing object). If a route becomes unavailable, AA-SBC removes it from the table. For example, if an interface becomes unavailable, AA-SBC removes all associated routes.

AA-SBC automatically installs (and the routing table displays) the loopback address. AA-SBC uses this address, with a metric of 0 and a name of <lo>, for its internal connections.

This route table is for management traffic, not SIP or media traffic. As a result, no load balancing occurs. If the table should have two equal cost routes to reach a destination, AA-SBC always uses the first listed route. Routes are ordered in the table by the length of the mask, since more specific routes are preferred.

Sample Output

```
NNOS-E> show routing
destination        type      gateway          source-ip       metric
   name
-----------        ----      -------          ---------       ------
   ----
192.168.0.1/32     host      0.0.0.0          192.168.0.1     1
   <eth0>
10.1.34.160/32     host      0.0.0.0          10.1.34.160     1
   <vx112>
172.26.0.155/32    host      0.0.0.0          172.26.0.155    1
   <vx111>
192.168.0.0/30     interface 0.0.0.0          192.168.0.1     1
   <eth0>
10.1.34.0/24       interface 0.0.0.0          10.1.34.160     1
   <vx112>
172.26.0.0/24      interface 0.0.0.0          172.26.0.155    1
   <vx111>
127.0.0.1/8        interface 0.0.0.0          127.0.0.1       0
   <lo>
172.0.0.0/8        network   172.26.0.254     172.26.0.155    1
   172 nets
0.0.0.0/0          default   10.1.34.254      10.1.34.160     1
   default
```

Properties

Field	Description
destination	The destination address, which the system resolves to the corresponding next-hop for the route. If the next-hop is on the same local network, the destination itself is the next hop.
type	The type of device the route represents, either: • **host**— a host IP address. This could come from either static or IP interface configuration. • **interface**— a network route that is directly connected via an interface. • **network**— an IP address and mask to match the destination network. Configured statically, this represents a network of hosts. • **default**—always at represented as 0.0.0.0/0, this route is applied for anything that has no other match in the table. Statically configure the gateway.
gateway	The next hop address for the route. A value of 0.0.0.0 indicates that the next hop is the destination itself.
source-ip	The IP address of the local interface on which the route is configured (the local interface address assigned with the route). When the route is selected this is the address the system uses to source the packet.
metric	The cost associated with the route, assigned when the route was configured (via ip or routing objects). The lower the metric the more preferred the route. The system chooses the more preferred route when there are multiple interfaces available on the same network.
name	The name assigned to the route. If the name indicates an ethernet (eth*X*) or virtual (vx*X*) interface, it is the route associated with the local interface on the box. Otherwise, it displays the name assigned when you created a static route with the routing object.

show rtp-transcode-info

Purpose

Displays session information relative to the transcoding of media types. Transcoding is the process of converting media from one CODEC into a different CODEC. This allows, in some cases, endpoints supporting different media types to communicate. See the media object for more information.

This command provides a quick snapshot of the current volume of transcoding activity, in terms of the number of streams, by state (active, DTMF-only, provisional, or unused state). It can be useful when troubleshooting log messages which indicate that transcoding limits have been exceeded.

Sample Output

```
NNOS-E> show rtp-transcode-info

          active: 2
       dtmf-only: 0
     provisional: 0
          unused: 0
   total-current: 2
         maximum: 2
exceeded-current: 0
  exceeded-count: 0
exceeded-seconds: 0
  exceeded-start:
    exceeded-end:
```

Properties

Field	Description
active	The number of rtp-transcode streams passing RTP packets.
dtmf-only	The number of rtp-transcode streams waiting only to decode DTMF.
provisional	The number of rtp-transcode streams waiting for an SDP answer.

Field	Description
unused	The number of rtp-transcode streams with not encoders or decoders, typically the other side of DTMF-only.
total-current	The current number of rtp-transcode streams.
maximum	The maximum number of concurrent rtp-transcode streams.
exceeded-current	The number of rtp-transcode streams currently above the threshold.
exceeded-count	The number of times the system has exceeded the threshold.
exceeded-seconds	The number of seconds the system has operated above the threshold.
exceeded-start	The time at which the system last went above the threshold.
exceeded-end	The time at which the system last went below the threshold.

show rtp-transcode-stats

Purpose

Displays statistics for active RTP transcoding sessions. It displays for each session ID the associated call leg, CODEC, and transcoding action taken by AA-SBC, as well as packet counts. Use this status provider to view active and summary statistics for RTP transcoding. See Transcoding media types for more information on the action taken by AA-SBC.

These transcoding statistics allow you to examine the RTP stream to determine which CODECs are being used for a given session. This command is most useful when trying to diagnose audio problems when transcoding is involved. Note that for basic calls, the rows marked with call-leg=1 describe the flow of media packets from the calling phone towards the answering phone (the forward path). The rows marked with call-leg=2 describe the flow of media from the answering phone towards the calling phone (the reverse path)

Sample Output

```
NNOS-E> show rtp-transcode-stats

session-id          call-leg  Action   Codec   payload-type  Packets
   Dropped

----------          --------  ------   -----   ------------  -------
   -------

0x4c2b7ee991990f5  1         Decode   iLBC    99            481
   0

                              Encode   pcma     8            721
   0

                    2         Decode   pcma     8            720
   0

                              Encode   iLBC    97            480
   0
```

Properties

Field	Description
session-id	The ID of the active RTP transcoding session.
call-leg	An index for a call leg indicating the direction of the call.
Action	The action the system takes on RTP packets, either Encode or Decode, on a portion of a call leg. In the first line of the sample output, the system decoded 481 iLBC packets to linear samples. It encoded those samples to 721 PCMA packets. The difference in packet count is a result of the difference in the PTime defaults of the CODECs. There may be multiple decode CODECs per leg (e.g., if the phone negotiates more than one), but there will always be only one encode CODEC per leg.
Codec	The name of the CODEC.

Field	Description
payload-type	A value negotiated in the SDP, indicating the CODEC type for each RTP packet. There are certain well-known values (values below 96), but for dynamic payload types (above 96), there must be an “rtpmap” in the SDP. When you add a CODEC to the SDP (using the media transcode-media-types property), the system adds the corresponding rtpmap.
Packets	The total number of packets processed for this CODEC type in this media stream.
Dropped	The total number of packets dropped for this CODEC type in this media stream.

show rtp-transcode-summary

Purpose

Displays summary statistics for RTP transcoding by CODEC. It includes actions and running packet counts since the last reboot. Use this status provider to view summary statistics for RTP transcoding. See Transcoding media types for more information on the action taken by AA-SBC.

This command is most useful when trying to determine if transcode policy elements are set up correctly, for example, whether the right number of CODEC encoders have been allocated.

Sample Output

```
NNOS-E> show rtp-transcode-summary

Codec    Action    Alloc    Used    Active    Packets    Dropped

-----    ------    -----    ----    ------    -------    -------

iLBC     Decode    1        1       1         884        0
         Encode    1        1       1         882        0

pcma     Decode    1        1       1         1323       0
         Encode    1        1       1         1326       0
```

Properties

Field	Description
Codec	The name of the CODEC for which statistics are being reported.
Action	The transcoding action being reported on, either encode or decode.
Alloc	The number of times the system added a CODEC to the SDP for potential encoding or decoding use.
Used	The total number times the system used the CODEC for encoding or decoding (a CODEC must have processed at least one packet to be counted).
Active	The total number of this CODEC type currently in use, for encoding and for decoding, although it may not have yet been used. It may be allocated and listed as available, but may not have yet processed and RTP packets.
Packets	The total number of packets received and sent, encoded and decoded, for this CODEC type.
Dropped	The total number of packets dropped, encoded and decoded, for this CODEC type.

show rules

Purpose

Displays the status and activity level of each rule configured in each policy. Rules are configured using the rule object.

Sample Output

```
NNOS-E> show rules

       name: policy default/rule regauth
      admin: enabled
evaluations: 0
  successes: 0

       name: policy default/rule abcTOxyz
      admin: enabled
evaluations: 0
  successes: 0

       name: policy default/rule abcTOdef
      admin: enabled
evaluations: 0
  successes: 0

       name: policy default/rule abc
      admin: enabled
evaluations: 0
  successes: 0
```

Properties

Field	Description
name	The name of the policy and the rule to which the policy applies.
admin	The administrative state of the rule associated with the named policy. If disabled, all other enabled rules under the named policy will still be checked against incoming SIP messages.
evaluations	The number of times the rule was applied against a SIP REQUEST message.
successes	The number of times a SIP REQUEST message matched the policy rule.

show sensor-events

Purpose

Displays events sent to the Intelligent Platform Management Interface (IPMI) event log. Output indicates the effected sensor and an event description and time stamp.

Sample Output

```
NNOS-E> show sensor-events
timestamp                 sensor             type                       d
   escription
```

```
---------                 ------            ----                     -
   ----------
11:48:31 Mon 2006-10-09   Logging
   Disabled0  event-logging-disabled   log area reset/cleared
11:48:38 Mon 2006-10-09   System
   Event0      system-event             OEM system boot event
111:48:50 Mon 2006-10-09  SATA  Drv 3
   Pres0  drive-slot               device removed/absent
11:48:55 Mon 2006-10-09   SATA  Drv 4
   Pres0  drive-slot               device removed/absent
11:48:55 Mon 2006-10-09   Power
   Redundancy0  power-unit               fully redundant
11:48:58 Mon 2006-10-09   POST
   Error0        system-firmware-progress error
11:48:58 Mon 2006-10-09   POST
   Error0        system-firmware-progress error
11:49:24 Mon 2006-10-09   System
   Event0      system-event             OEM system boot event
11:49:24 Mon 2006-10-09   System
   Event0      system-event             timestamp clock sync
11:02:10 Wed 2006-10-18   SATA  Drv 6
   Pres0  drive-slot               device removed/absent
11:02:13 Wed 2006-10-18   Power
   Redundancy0  power-unit               fully redundant
```

Properties

Field	Description
timestamp	The time the sensor event occurred.
sensor	The name of the effected sensor.
type	The category into which the sensor falls.
description	Descriptive text that indicates the specific action that triggered the event.

show sensor-info

Purpose

Displays version and state information for the Intelligent Platform Management Interface (IPMI).

Sample Output

```
NNOS-E> show sensor-info
  version: 2.0
    state: up
   faults:
self-test: Success!
```

Properties

Field	Description
version	The version of firmware running on the interface.
state	The state of the IPMI, either: • initializing—initializing IPMI • unsupported—IPMI is not supported on this platform • up—IPMI is active
faults	A reading of any detected hardware faults. The field is left blank if there are no faults, or displays one of the following: • controller—the controller failed to power the system on or off. • power—there was a main power subsystem fault. • interlock—the chassis power interlock switch is active. • overload—there is a power overload. • fan—a cooling fan fault has been detected • drive—there is a drive fault. • intrusion—there has been a chassis intrusion.
self-test	The results of the IPMI power-on self-test. If you receive anything other than Success!, contact Technical Support.

show sensors

Purpose

Displays status information for the various system hardware sensors (temperature, fan speed, etc.)

Sample Output

```
NNOS-E> show sensors
sensor                   type              value
------                   ----              -----
+1.5V NIC Core           voltage           1.5288 volts
AUX +3.3V                voltage           3.264 volts
BB +1.2V Vtt             voltage           1.2096 volts
BB +1.5v                 voltage           1.5132 volts
BB +12V                  voltage           12.152 volts
BB +3.3V                 voltage           3.354 volts
BB +5V                   voltage           5.07 volts
BB -12V                  voltage           -11.758 volts
BB Vbat0                 battery           001------------
BMC Watchdog0            watchdog-2        0000----0------
Baseboard Fan 1          fan               5254 RPM
--more--
```

Properties

Field	Description
sensor	The name of the effected sensor.
type	The category into which the sensor falls.
value	The current value or reading of the associated sensor.

show server-conn-lookup

Purpose

Displays entries of the SIP host lookup table that are using either TCP or TLS for transport and have a **host** configured. These entries are the servers configured using the server-pool server object or the gateways configured with the switch object. Use the show server-host-lookup command to display entries with hosts using any transport protocol.

Sample Output

```
NNOS-E> show server-conn-lookup
peer-name   server-name   trunk-name   connection  host        TPT   l
   ocal  hits
---------   -----------   ----------   ----------  ----        ---   -
   ----  ----

Cluster     10.1.34.13    10.1.34.13   0           10.1.34.13  TLS   5
   061   294
```

Properties

Field	Description
peer-name	SIP gateway server or carrier name.
server-name	The name of the server-pool server or gateway.
trunk-name	The name of a trunk group associated with the corresponding carrier gateway. If no gateway is configured, or the peer is a server and not a carrier, the field is blank.
connection	The memory address of the socket hosting the TCP or TLS connection.
host	The host name or IP address of the server or gateway involved in the connection. This is derived from the **host** field of the server or carrier configuration.
TPT	The transport protocol in use for the connection with this server, either TCP or TLS.

Field	Description
local	The port number the system is configured to use (with the server **local-port** property) in the Contact header, Via header, and source port when it sends a Register request (and subsequent SIP messages) to an upstream server. The server caches the binding and includes the local-port when contacting the system. Additionally, the server can be configured to send SIP messages to this particular local-port without prior registration from the system. With local-port configured, the system can tell: • to which connection in the server pool to forward a call. • which connection in the server pool it received the call from, when the connection sends SIP message to this local port.
hits	The number of times the system has forwarded traffic to the server.

show server-host-lookup

Purpose

Displays entries of the SIP host lookup table that are using any transport protocol and have a **host** configured. These entries are the servers configured using the server-pool server object or the gateways configured with the switch object. Use the show server-conn-lookup command to display only those entries using TCP or TLS for transport.

Sample Output

```
NNOS-E> show server-host-lookup
peer-name   server-name   trunk-name   connection  host        TPT   l
   ocal  hits
---------   -----------   ----------   ----------  ----        ---   -
   ----  ----

Cluster     10.1.34.13    1.1.1.1      0           1.1.1.1     UDP   0
         294
NNOS-E@NY      NY1                        0           100.0.0.1   TLS
     15061  5693
NNOS-E@SSJ     SJ1                        0           200.0.0.1   TLS
     25061  6923
```

Properties

Field	Description
peer-name	The SIP gateway server or carrier name.
server-name	The name of the server-pool server or gateway.
trunk-name	The name of a trunk group associated with the corresponding carrier gateway. If no gateway is configured, or the peer is a server and not a carrier, the field is blank.
connection	The memory address of the socket hosting the TCP or TLS connection.
host	The host name or IP address of the server or gateway involved in the connection. This is derived from the **host** field of the server or carrier configuration.

Field	Description
TPT	The transport protocol in use for the connection with this server, either TCP, TLS, or UDP.
local	The port number the system is configured to use (with the server **local-port** property) in the Contact header, Via header, and source port when it sends a Register request (and subsequent SIP messages) to an upstream server. The server caches the binding and includes the local-port when contacting the system. Additionally, the server can be configured to send SIP messages to this particular local-port without prior registration from the system. With local-port configured, the system can tell: • to which connection in the server pool to forward a call. • which connection in the server pool it received the call from, when the connection sends SIP message to this local port.
hits	The number of times the system has forwarded traffic to the server.

show services-routing

Purpose

Displays a summary of the routes in all service routing tables. Information displayed includes the table type, destination and gateway, source IP address and the origin.

The verbose form displays services-routing metrics and cluster media-partners information (for media). In the CLI, you can filter the output can to display a specific table by entering the name of the service route table to display in the command line (e.g., show services-routing media).

The origin is either local or cluster, where local is a local route on the box itself and cluster means the route was learned from another box in the cluster. Source IP is the local interface to reach that route destination. If the route is a local route, then the source-ip is the local IP interface on which the route was configured. If the route is a cluster route, the source-ip is the local interface used to communicate with other cluster box that advertised the route (i.e., the local interface in which the route was learned).

Sample Output

```
NNOS-E> show services-routing
service                destination         gateway          source-ip        origin
-------                -----------         -------          ---------        ------
sip                    10.1.34.0/24        0.0.0.0          10.1.34.160      local
sip                    172.26.0.0/24       0.0.0.0          172.26.0.155     local
sip                    172.0.0.0/8         172.26.0.254     172.26.0.155     local
sip                    0.0.0.0/0           10.1.34.254      10.1.34.160      local
media                  10.1.34.0/24        0.0.0.0          10.1.34.160      local
media                  172.26.0.0/24       0.0.0.0          172.26.0.155     local
media                  172.0.0.0/8         172.26.0.254     172.26.0.155     local
media                  0.0.0.0/0           10.1.34.254      10.1.34.160      local
```

Properties

Field	Description
service	The name of the service, either media, stun, or sip.
destination	The destination address, which the system resolves to the corresponding next-hop for the route. If the next-hop is on the same local network, the destination itself is the next hop.
gateway	The next hop address for the route. A value of 0.0.0.0 indicates that the next hop is the destination itself.

Field	Description
source-ip	The local interface to reach the route destination. If the route is a local route, then the source-ip is the local IP interface on which the route was configured. If the route is a cluster route, the source-ip is the local interface used to communicate with the other system cluster box that advertised the route (i.e., the local interface from which the route was learned).
origin	The origin of the route, either local or cluster. Local means a local route on the box itself and cluster means the route was learned from another box in the cluster.

show services-routing-tables

Purpose

Displays all the services routing tables (the default tables and those created as a result of tag-based route selection configured on an ip interface). The display also includes a total route count, which includes both active and inactive routes.

```
NNOS-E> show services-routing-tables

service                      route-count
-------                      -----------
media                        3
media.qik-eastmode           3
media.tomp                   3
sip                          3
sip.lcs1-eastmode            3
sip.qik-eastmode             3
sip.tomp                     3
stun                         0
```

Properties

Field	Description
service	The name of the services routing table.
route-count	The number of routes in the table.

show sip-authentication

Purpose

Displays information about SIP authentication messages handled by the AA-SBC.

Sample Output

```
NNOS-E>show sip-authentication

                                      name: default
                sip-stack-pre-auth-timeout: 30 seconds
           sip-stack-pre-auth-max-pendings: 1024 seconds
    total-blocking-authentication-messages: 0
         total-sip-stack-pre-auth-messages: 0
            total-auth-suppressed-messages: 0
  total-sip-stack-pre-auth-api-timeouts: 0 seconds
         total-sip-stack-pre-auth-timeouts: 0
total-sip-stack-pre-auth-unmatched-replies: 0
           total-sip-stack-pre-auth-queued: 0
            most-sip-stack-pre-auth-queued: 0
```

Properties

Field	Description
name	The name of the VSP instance.
sip-stack-pre-auth-timeout	The number of seconds before the server timesout and the discards the message.
sip-stack-pre-auth-max-pendings	The number of seconds the allows a message to stay in its queue pending authentication.
total-blocking-authentication-messages	The number of authentication messages that will cause the to block message processing.
total-sip-stack-pre-auth-messages	The number of messages the has authorized.
total-auth-suppressed-messages	The number of authentication messages that have been suppressed. An authentication message is suppressed if the receives a REGISTER for an AOR that has not yet expired. Rather than authenticating, the sends back a 200 OK.

Field	Description
total-sip-stack-pre-auth-api-timeouts	The number of times the sends a message to the internal Authentication process and does not receive a response.
total-sip-stack-pre-auth-timeouts	The number of times an authentication message expired while waiting in the queue.
total-sip-stack-pre-auth-unmatched-replies	The number of authentication messages that failed because the could not find a match for authorization credentials.
total-sip-stack-pre-auth-queued	The total number of queued authorization messages.
most-sip-tack-pre-auth-queued	The Maximum number of queued authentication messages allowed.

show sip-authorization-details

Purpose

Displays detailed information about SIP authorization on the .

Sample Output

```
NNOS-E>show sip-authorization-details
--------------------------------------------------------------------------------
Provider      Requests   Accepts  Average   Errors  Average  QClipped    Others
--------------------------------------------------------------------------------
Local                0         0    0.000        0    0.000         0         0
WSDL                 0         0    0.000        0    0.000         0         0
Diameter             0         0    0.000        0    0.000         0         0
RADIUS               0         0    0.000        0    0.000         0         0
--------------------------------------------------------------------------------
```

Properties

Field	Description
Provider	The protocol to be used for authorization.
Requests	The number of requests submitted to each provider.
Accepts	The number of positive replies received from the remote server.
Average	The average response time, in seconds, for each Accept.
Errors	The number of errors (rejected by the server, timed out, etc.).
Average	The average response time, in seconds, for each Error response. Note that this value is maintained separately because a server often dealsy a negative response for 1-2 seconds in an attempt to avoid attacks.

Field	Description
QClipped	The number of requests that failed locally, without ever being sent to the remote server, because the queue of requests outstanding to the server(s) has grown too long. This is often indicative of problems.
Others	The sum of any other types of errors. These can be seen individually by adding **-v** to the end of this action.

show sip-peers

Purpose

Displays the name, domain, type and failover detection method of all configured SIP registration peers. Peers are identified with the server object.

Sample Output

```
NNOS-E> show sip-peers
name                    domain           type        mode        detect
----                    ------           ----        ----        ------
Cluster 10.1.34.13      vfn.com          provider    provider    none
```

Properties

Field	Description
name	The name of the configured enterprise SIP server.
domain	The domain name associated with the server.
type	The service-type setting for the server, either provider, internal, or external. Service type specifies the way in which the system handles INVITE and REGISTER requests and database exchanges.

Field	Description
mode	The status of the peer. When a server is down (not reachable), if the **routing-setting** property of the pstn-backup attribute is not selected, the system changes the state of the server to “not available.” If it is selected, an unavailable server's state changes to “local mode.” In its normal state, the system operates in provider mode, forwarding calls to a provider's application server. If the server fails, and the system has location information for the provider, it forwards calls locally. Otherwise, the system forwards calls to a PSTN gateway. You configure the gateway using the pstn-gateway server object. This is called local mode.
detect	The **failover-detection** setting for the server. Failover detection determines the method to use to detect a when a upstream server peer is unavailable. This field could display none, auto, ping, or register.

show sip-server-availability

Purpose

Displays server-pool server-pool-admission-control or carrier switch configuration and status information for each configured SIP server. Configuration data includes transport protocol, port, failover detection method, counts, and thresholds.

Sample Output

```
NNOS-E> show sip-server-availability
host        transport  port  detect waiting time count threshold fallb
    ack status
----        ---------  ----  ------ ------- ---- ----- --------- -----
    --- ------
1.1.1.1     UDP        5060  none   false   0    0     4         300
        up
2.2.2.2     UDP        5060  none   false   0    0     4         300
        up
10.1.34.13  TLS        5061  none   false   0    0     4         300
        up
```

Properties

Field	Description
host	The name of IP address of the server-pool-admission-control or switch.
transport	The protocol used by the server-pool-admission-control or switch, either any, TCP, UDP, or TLS.
port	The port used by the server-pool-admission-control or switch for SIP traffic.
detect	The **failover-detection** setting for the hosting server or carrier, either none, auto, ping, or register. Failover detection determines the method to use to detect a when a upstream server peer is unavailable.
waiting	The state of the system pinging the host. If true, the system has sent a ping and is awaiting a response. If false, it has not pinged the host.
time	The amount of time (number of seconds) the system has waited for a response to a ping to the host.
count	The number of failed pings to the host (the dead count).
threshold	The **dead-threshold** setting for the hosting server or carrier. This threshold specifies the number of transaction failures (and resulting retransmissions) a server can experience before the server state is changed to DOWN.
fallback	The **dead-fallback-interval** setting for the hosting server or carrier. During this period, the system does not send REGISTER or INVITES to the down server.
status	The current status of the server-pool server-pool-admission-control or carrier switch.

show sip-server-pool

Purpose

Displays all hosts available for each server-pool server-pool-admission-control peer, and the status of each. In addition, the output indicates configuration settings, and the number of requests sent to the peer.

Sample Output

```
NNOS-E> show sip-server-pool
peer-name     server        host         TRPT   port   box    state  h
   its  pref
---------     ------        ----         ----   ----   ---    -----  -
   ---  ----
PBX Maynard   PBX
   Maynard   0.0.0.0      UDP    5060   local  up     0     none
PBX Boston    PBX
   Boston    0.0.0.0      UDP    5060   local  up     0     none
NNOS-E@SanJose   SJ1           200.0.0.1    TLS    5061   1      up
     0     100
NNOS-E@SanJose   SJ2           200.0.0.2    TLS    5061   1      up
     0     200
```

Properties

Field	Description
peer-name	The name of the server hosting the connection. This is the enterprise server.
server	The name of the server entry in the server pool, configured with the server-pool server-pool-admission-control object. If the enterprise server is of type **sip-connection**, the server is the same as the peer-name.

Field	Description
host	The host address of the server-pool server-pool-admission-control. This value is configured with the **host** property. A host address of 0.0.0.0 indicates that the host is not configured. For a server of type **sip-connection**, a host of 0.0.0.0 indicates that the host will be learned dynamically through registrations. To do so, however, the local-port property must be set (non-zero).
TRPT	The protocol used by the connection.
port	The port used by the connection for SIP traffic.
box	The box on which the server is configured. A value of 0 indicates the local box.
state	The operational state of the hosting server. This state is determined by failover checks, configured in the enterprise server **failover-detection** property. If that property is set to **none**, the state always appears as up.
hits	The number of requests sent to the peer.
pref	The preference assigned to the server-pool server, which specifies the preference for the connection. The lower the value the higher the preference. If you use the value of **none**, the system uses the preference set in a different part of the configuration, such as the ordered set of arbitration rules in the dial-plan object.

show sip-stack

Purpose

Displays general statistics about the SIP process. (Other SIP display commands provide more detailed counters specific to an aspect or process.)

Sample Output

```
NNOS-E> show sip-stack
                                 name: default
                                 mode: auto-determine
```

```
                           interfaces: 0
                         active-calls: 1
                      connected-calls: 1
                          total-calls: 10
                   total-failed-calls: 7
                  active-associations: 5
                      active-sessions: 1
               total-message-received: 455
                   total-message-sent: 507
           total-dns-pending-messages: 0
          total-enum-pending-messages: 0
      total-location-pending-messages: 0
total-authentication-pending-messages: 0
                       worker-threads: 20
                       socket-threads: 1
                               status: Running
                         policy-epoch: 0
               call-admission-control: enabled
                            max-calls: 1000
                    max-registrations: 30000
                   max-calls-in-setup: 200
                        current-calls: 1
               current-calls-in-setup: 0
                    max-calls-dropped: 0
           max-calls-in-setup-dropped: 0
                        max-tls-calls: 10000
               max-tls-calls-in-setup: 500
                    current-tls-calls: 0
           current-tls-calls-in-setup: 0
                max-tls-calls-dropped: 0
       max-tls-calls-in-setup-dropped: 0
```

Properties

Field	Description
name	The name of the VSP that this command is reporting on.
mode	The SIP operating mode to use with this server, as configured with the sip-settings object of the session config. Settings are either: • auto-determine—the system determines the mode. • proxy—the system is the SIP proxy that provides SIP registration, location, policy, and other services that determine the outcome of the SIP call.
interfaces	The number of active SIP interfaces on the local box.
active-calls	The total number of calls, both in progress and connected, that the system is currently handling.
connected-calls	The number of connected calls currently being managed by the system.
total-calls	The total number of calls processed by the system since the last boot.
total-failed-calls	The total number of calls that did not make it to the connect state since the last boot.
active-associations	The number of “address pairs” that were used for communications through the system since it last boot. When a session is setup, there is a TO and FROM user (e.g., joe@123.com calls bob@456.org). The association is a unique identifier assigned to that combination.
active-sessions	The number of currently active SIP sessions. Active sessions include, calls, registrations, and presence sessions.
total-message-received	The total number of SIP messages received since last boot.
total-message-sent	The total number of SIP messages sent since last boot.

Field	Description
total-dns-pending-messages	The total number of SIP messages awaiting DNS lookup.
total-enum-pending-message	The total number of SIP messages awaiting ENUM lookup.
total-location-pending-messages	The total number of SIP messages awaiting master location database lookup.
total-authentication-pending-messages	The total number of SIP messages awaiting authentication verification.
worker-threads	The value of the **stack-worker-threads-max** property set in the **settings** object. This value is the number of SIP stack processing threads to create for this VSP. If the value displays as 0, the system executes a single thread.
socket-threads	The value of the **stack-socket-threads-max** property set in the **settings** object. This value is the number of SIP stack processing threads that should be used for TLS processing. If you are not using TLS, the value should be 1. If you are using TLS, value should be 4.
status	The state of the SIP stack, either running or disabled. This can be administratively altered using the **admin** property of the **vsp** object.
policy-epoch	A general counter that indicates the number of times there has been a change to a session-config, default-session-config, rule, enterprise server, or server-pool server configuration. This value is used internally for policy update decisions.
call-admission-control	The call-admission control setting. This setting must be enabled for the following to be applicable: • max-calls-in-setup • max-number-of-tls • max-tls-in-setup If disabled, only the more general **max-number-of-sessions** property controls setup and connection limits.

Field	Description
max-calls	The maximum number of concurrent SIP sessions that the VSP can support. This value includes all REGISTER, SUBSCRIBE, INVITE, and other sessions. This value is set with the **max-number-of-sessions** property in the vsp>**settings** object
max-registrations	The maximum number of registrations allowed in the location cache. When the system reaches the maximum registration count, registrations are denied until the number falls below this threshold. This value is set with the **max-number-of-registrations** property in the vsp>**settings** object
max-calls-in-setup	The value of the **max-calls-in-setup** property set in the **vsp** object. This setting controls the maximum number of inbound call legs in setup stage allowed by the CAC.
current-calls	The current number of SIP sessions.
current-calls-in-setup	The current number of SIP sessions in setup stage.
max-calls-dropped	The number of SIP sessions that were dropped because you hit the **max-calls** threshold.
max-calls-in-setup-dropped	The number of in-progress SIP sessions that were dropped because you hit the **max-calls-in-setup** threshold.
max-tls-calls	The value of the **max-number-of-tls** property set in the **vsp** object. This setting controls the maximum number of TLS connections allowed on the VSP. This would include TLS connections for any type of SIP traffic, and includes TLS calls in setup and those that are established.
max-tls-calls-in-setup	The value of the **max-tls-in-setup** property set in the **vsp** object. This setting controls the maximum number of TLS connections allowed to be in the setup stage at one time. Establishing a TLS connection is very compute-intensive, so this value helps protect the system from being over-burdened by TLS connections.

Field	Description
current-tls-calls	The total number of TLS connections currently handled by the system, both established and in setup.
current-tls-calls-in-setup	The number of TLS connections currently in setup.
max-tls-calls-dropped	The number of TLS connections that were dropped because you hit the **max-tls-calls** threshold.
max-tls-calls-in-setup-dropped	The number of in-progress TLS connections that were dropped because you hit the **max-tls-calls-in-setup** threshold.

show sip-trunk-ports

Purpose

Displays the bindings of AORs to trunk ports. The configuration for this is a result of the registration-plan route **alter-contact** property set to **trunk-port-per-binding**.

Trunk ports are allocated based on AORs, and the AORs are sent in the phone registrations. This command is most useful when registrations are failing because it allows you to observe the trunk ports. For example, if a registration fails, you can compare the allocated ports to the ports specified in the SIP message to determine whether there is a problem. In other cases, if SIP messages (like INVITES or NOTIFYs) are not being responded to correctly, it could be the result of a SIP message specifying an AOR/trunk port combination that has not been properly allocated.

The output of this command is most useful when combined with other information, such as SIP message logs and code traces.

Sample Output

```
NNOS-E> show sip-trunk-ports

outbound-port         source-port         owner                    hits
-------------         -----------         -----                    ----
172.26.0.109:24477    172.30.0.177:7584   sip:7812454444@rk.com 3
```

Properties

Field	Description
outbound-port	The IP address and port number of the system.
source-port	The IP address and port number of the client server.
owner	The AOR of the source.
hits	The number of times the system was able to match the call to the port allocated for the caller (source).

show system-info

Purpose

Displays system information, including operating system release and version information and other integrated software information.

Sample Output

```
NNOS-E> show system-info
Machine type:      i686
System name:       Linux
Node name:         172.26.0.155
Box identifier:    01a1-6f9e-dcfa-9b9e
Kernel version:    2.6.11-4-cov
OS version:        1.4-22573-b3.2.0
OS uptime:         1 days 19:52:58
LIBC version:      glibc 2.3.4
Pthread version:   NPTL 2.3.4
OpenSSL version:   OpenSSL 0.9.8c 26 Sep 2006 (companyXYZ-20291)
  Export status:   Full version
Database version:  07.02.0005 PostgreSQL 8.1.2
```

Properties

Field	Description
Machine type	The Intel processor type.
System name	The system's operating system.
Node name	The host name configured under the **box** object. If no host name is configured, the system displays the default host name, micro4.
Box identifier	This is a box identifier, used for licensing purposes. It is assigned during the manufacturing process.
Kernel version	The version of the kernel that is currently running on the system. Use the **show kernel-version** command for more details about the kernel build.
OS version	The version of the operating system running on the system.
OS uptime	The length of time since the last cold start.

Field	Description
LIBC version	The version of the GNU C Library currently running on the system.
Pthread version	The version of the POSIX Threads API, the standard for creating and manipulating threads, currently running on the system.
OpenSSL version	The version of OpenSSL, used for SSL and TLS connections, currently running on the system.
Export status	The cipher shipping status, either full version or export version. The export version does not contain the complete suite of TLS ciphers.
Database version	The version of the software used for managing all system databases currently running on the system.

show tcp

Purpose

Displays local and remote TCP session state information, such as ESTABLISHED and LISTEN. supports two types of ports—listeners and connections.

Sample Output

```
NNOS-E> show tcp
local                 remote                state
-----                 ------                -----
10.1.34.160:5060      0.0.0.0:0             LISTEN
10.1.34.160:5061      0.0.0.0:0             LISTEN
10.1.34.160:33617     10.1.34.13:5061       SYN-SENT
127.0.0.1:5432        0.0.0.0:0             LISTEN
172.26.0.155:22       0.0.0.0:0             LISTEN
172.26.0.155:22       172.26.3.63:3296      ESTABLISHED
172.26.0.155:22       172.30.1.2:4474       ESTABLISHED
172.26.0.155:80       0.0.0.0:0             LISTEN
172.26.0.155:5060     0.0.0.0:0             LISTEN
192.168.0.1:5132      192.168.0.2:33057     ESTABLISHED
```

Properties

Field	Description
local	If the system is: • **listener**—the IP address on the system used for the TCP connection. • **initiator**—the system interface from which the connection originated.
remote	If the system is: • **listener**—the remote IP address is 0.0.0.0 until a system connects to the listener. At that point, the remote IP becomes the address of the endstation connecting. • **initiator**—the IP address of the endstation the system is connecting to.
state	The state of the TCP connection. See the following table for a description of each TCP state.

The following table describes each of the TCP states, as defined in *RFC 793, Transmission Control Protocol Specification*:

State	Indicates...
LISTEN	waiting for a connection request from any remote TCP and port.
SYN-SENT	waiting for a matching connection request after having sent a connection request.
SYN-RECEIVED	waiting for a confirming connection request acknowledgment after having both received and sent a connection request.
ESTABLISHED	an open connection, data received can be delivered to the user. The normal state for the data transfer phase of the connection.
FIN-WAIT-1	waiting for a connection termination request from the remote TCP, or an acknowledgment of the connection termination request previously sent.
FIN-WAIT-2	waiting for a connection termination request from the remote TCP.

State	Indicates...
CLOSE-WAIT	waiting for a connection termination request from the local user.
CLOSING	waiting for a connection termination request acknowledgment from the remote TCP.
LAST-ACK	waiting for an acknowledgment of the connection termination request previously sent to the remote TCP (which includes an acknowledgment of its connection termination request).
TIME-WAIT	waiting for enough time to pass to be sure the remote TCP received the acknowledgment of its connection termination request.
CLOSED	no connection state at all (and therefore never seen).

show timezones

Purpose

Displays a list of all preconfigured time zones recognized by and their associated codes. (There are in excess of 1700 definitions known by .) A system must be set to the time zone in which it is located so that time stamps and settings are correct in the software. If you type set timezone ? from the box object, the system displays about 30 of the most popular time zones. To display the code for a time zone other than one of the more common ones, use this command.

Sample Output

```
NNOS-E> show timezones

name                             codes
----                             -----
Africa/Abidjan                   LMT GMT
Africa/Accra                     LMT GHST GMT
Africa/Addis_Ababa               ADMT EAT
Africa/Algiers                   PMT WEST WET CEST CET
Africa/Asmara                    ADMT EAT
Africa/Asmera                    ADMT EAT
Africa/Bamako                    LMT GMT WAT
--More--
```

Properties

Field	Description
name	The geographic location of the time zone.
codes	The time zone abbreviation string.

show trap-categories

Purpose

Displays the reported SNMP traps and their associated categories. You can then filter the SNMP traps by categories. The filter determines which categories of SNMP traps sends out the WSDL interface to the external event server. Set the filter with the event-group object. The complete list of SNMP traps are listed in Appendix A of *Net-Net OS-E – Using the NNOS-E Management Tools*.

Sample Output

```
NNOS-E> show trap-categories

category   trap
--------   ----
csta       CallConnected
csta       CallCreated
csta       CallHeld
csta       CallRetrieved
csta       CallTerminated
csta       PlayComplete
csta       RecordComplete
dos       DosSIPPolicyTrap
dos       DosTransportPolicyTrap
dos       DosUrlPolicyTrap
h323      H323CallAlerting
h323      H323CallConnected
h323      H323CallCreated
h323      H323CallDisconnected
--More--
```

Properties

Field	Description
category	The SNMP trap category used to filter the traps going out the system to the external event server.
trap	The name of the trap. For example, DosTransportPolicyTrap indicates that a dynamic policy rule was instituted in response to a Transport Policy threshold being crossed.

show version

Purpose

Displays version information for each application that makes up the software suite. This command is similar to the **show module-version** command, which displays for the kernel modules (the modules that are part of the operating system).

Sample Output

```
NNOS-E> show version
image    version    build        branch     time                         computer
-----    -------    -----        ------     ----                         --------
monitor  3.2.0      22831        b3.2.0     00:42:15 Sun 2006-12-17  AUTOBUILD
manager  3.2.0      22831        b3.2.0     00:43:10 Sun 2006-12-17  AUTOBUILD
SIP      3.2.0      22831        b3.2.0     00:58:29 Sun 2006-12-17  AUTOBUILD
media    3.2.0      22831        b3.2.0     00:59:40 Sun 2006-12-17  AUTOBUILD
reg      3.2.0      22831        b3.2.0     00:43:36 Sun 2006-12-17  AUTOBUILD
web      3.2.0      22831        b3.2.0     01:09:00 Sun 2006-12-17  autobuild
WS       3.2.0      22831        b3.2.0     01:12:00 Sun 2006-12-17  autobuild
```

Properties

Field	Description
image	The application name.
version	The software version installed on the system for the corresponding application.
build	The specific build of the software version for the corresponding application.
branch	The internal development tracking ID for the build.
time	The date and time of the build.
computer	The name of the computer system that built the image, typically AUTOBUILD.

show vrrp

Purpose

Displays state and configuration information for each configured VRRP vinterface.

Sample Output

```
NNOS-E> show vrrp
vrrp-interface: vx112
  vinterface-admin-status: enabled
    active-host-interface: eth2
               vrrp-state: Master
               vrouter-id: 113
          hosted-priority: 255
active-host-intf-op-state: up
               vrrp-group: 2
                    bound: true
               host-index: 0
     activation-timestamp: 10:56:29 Thu 2006-12-21
                 duration: 0 days 01:17:57
```

Properties

Field	Description
vrrp-interface	The name of the VRRP interface (the VX ID), configured with the vinterface object.
vinterface-admin-status	The administrative status of the virtual interface. When **enabled**, the referenced interfaces participate in the failover features of VRRP. When **disabled**, the interfaces do not serve as link backup.
active-host-interface	A referenced Ethernet interface used as the host for this VRRP interface. Only the current Master VRRP interface lists as the active host; this field is blank for all Backups.
vrrp-state	The role or state of the VRRP interface, either Init, Master or Backup. The role is established by your order of entry. The first host is Master, and backups are ordered according to their position. Use the **move** command to change the order. A state of Init indicates that the host interface is not configured.
vrouter-id	The vrouter ID, as required by the VRRP specification. This identifier is calculated by the system by adding one to the vinterface value. For example, if the vinterface is vx2, the vrouter-id is 3.
hosted-priority	The priority of the VRRP interface, as defined by the *RFC 2338, Virtual Router Redundancy Protocol*. The Master interface must have a priority of 255. Backup interfaces have a value of one less than the interface above them in the configuration order.
active-host-intf-op-state	The operational state of the Master.
vrrp-group	The VRRP group of which the interface is a member. Grouping interfaces is a configuration technique to apply failover. A vinterface with a group number of 0 does not participate in grouping.
bound	A indication of whether the interface is Master. The output should be **true** for a Master interface and **false** for Backups.
host-index	An internal identifier.

Field	Description
activation-timestamp	The time at which the VRRP interface became the active Master.
duration	The length of time that the VRRP interface has been active Master.

show vrrp-hosts

Purpose

Displays the status for each host configured for a VRRP interface. The information displayed includes active state, link status, priority, and failover timer setting for the vinterface **host-interface** property. You can have multiple local hosts for a vrrp interface (eth0, eth1, eth2, etc.); this status provider indicates the status of each host.

Sample Output

```
NNOS-E> show vrrp-hosts
         vrrp-interface: vx111
         host-interface: eth1
host-interface-op-state: up
                  bound: true
               priority: 255
   master-down-interval: 600
   activation-timestamp: 10:56:29 Thu 2006-12-21
               duration: 0 days 01:17:57

         vrrp-interface: vx112
         host-interface: eth2
host-interface-op-state: up
                  bound: true
               priority: 255
   master-down-interval: 600
   activation-timestamp: 10:56:29 Thu 2006-12-21
               duration: 0 days 01:17:57
```

Properties

Field	Description
vrrp-interface	The name of the VRRP interface (the VX ID), configured with the cluster>vrrp>**vinterface** object.
host-interface	A referenced Ethernet interface used as the host for this VRRP interface.
active-host-intf-op-state	The operational state of the Master.
bound	A indication of whether the interface is Master. The output should be **true** for a Master interface and **false** for Backups.
priority	The priority of the VRRP interface, as defined by the *RFC 2338, Virtual Router Redundancy Protocol*. The Master interface must have a priority of 255. Backup interfaces have a value of one less than the interface above them in the configuration order.
master-down-interval	A value calculated by the system and used to determine when the Master is determined to be down and the system should failover to the next configured interface.
activation-timestamp	The time at which the VRRP interface became the active Master.
duration	The length of time that the VRRP interface has been active Master.

show vx-bindings

Purpose

Displays the binding between a VX interface and its associated Ethernet interface. The binding is configured using the vinterface **host-interface** property.

Sample Output

```
NNOS-E> show vx-bindings
vx-interface    ethernet
------------    --------
```

```
vx0             eth4
vx1             eth5
vx2             eth4

NNOS-E> show vx-bindings
vx-interface    ethernet
------------    --------

vx0             unbound
vx1             unbound
vx2             unbound
```

Properties

.

Field	Description
vx-interface	The name of the VX interface (the VX ID), configured with the VRRP vinterface object.
ethernet	A referenced Ethernet interface used as the host for this VX interface. A value of **unbound** indicates that the vinterface is not associated with an Ethernet interface.

show web-services-callout-status

Purpose

Displays data for Web service requests made from to a PC, including the endpoint involved, the specific request, and the number of times the request occurred. The output also displays response time statistics. This command only appears as available when the web-service server is configured and enabled.

Sample Output

```
NNOS-E> show web-services-callout-status

      type: event
  endpoint: http://172.10.10.10:80/covws/callouts
      sent: 0
  min-sent: 0 ms
  avg-sent: 0 ms
  max-sent: 0 ms
    failed: 10
min-failed: 507 ms
avg-failed: 1045 ms
max-failed: 5509 ms
```

Properties

Field	Description
type	The type of callout the system made to the remote system, either: • event—the system sent an event. • policy—the system requested a policy. • location—the system requested a location.
endpoint	The URL of the remote web service that we sent the event to or requested information from.
sent	The number of successful transactions between the system and the web service.
min-sent	The fastest single transaction time between the system and a web service.
avg-sent	The average transaction time between the system and web services.
max-sent	The longest single transaction time between the system and a web service.
failed	The number of failed transactions between the system and the web service.
min-failed	The quickest time awaiting a failure response between the system and a web service.
avg-failed	The average failure response time between the system and a web service.
max-failed	The longest time awaiting a failure response between the system and a web service.

show web-services-client-status

Purpose

Displays configuration information for clients that have contacted a web service endpoint implemented by . This command only appears as available when the web-service server is configured and enabled. The output of this command displays one entry per client/endpoint combination. In the example below, client 172.30.0.210 contacted the templates service on nine times.

Sample Output

```
NNOS-E> show web-services-client-status
endpoint: /templates
  client: 172.30.0.210
   count: 9
```

Properties

Field	Description
endpoint	The web services endpoint on the system that was contacted.
client	The IP address of the client that contacted the system.
count	The number of requests that the system received from web service clients.

show web-services-request-status

Purpose

Displays data for Web service requests made from a remote computer to AA-SBC, including the endpoint involved, the endpoint service URL, and the number of times the request occurred. The output also displays response time statistics. This command only appears as available when the web-service server is configured and enabled. Note that the output displays an entry for each endpoint/function combination.

Sample Output

```
NNOS-E> show web-services-request-status

endpoint: /templates
 request: GetTemplate
   count: 8
     avg: 0 ms
     min: 0 ms
     max: 1 ms

endpoint: /templates
 request: GetTemplateFiles
   count: 1
     avg: 2 ms
     min: 2 ms
     max: 2 ms
```

Properties

Field	Description
endpoint	The web services endpoint on the system that was contacted.
request	The type of request from the remote computer
count	The number of requests, of the type identified in the **request** field, the endpoint has received.
avg	The average response time for all requests of this type to this endpoint. A response time under one millisecond reports as zero.
min	The fastest response to a request of this type to this endpoint.
max	The slowest response to a request of this type to this endpoint.

5. Access objects

Access description

The user access control defines the users that are allowed access to AA-SBC device and the specific privileges that they are granted. There are two access points within the system for granting privileges. You can assign access system-wide, providing access to the entire box. This is done from the top level of the configuration hierarchy. Or, you can configure access to a specific VSP. This is done through the VSP configuration object.

System-wide users log in with their user name:

```
username: jdoe
```

VSP users log in with the VSP name followed by their user name:

```
username: cxc1\jdoe
```

Whether from the top level or within a VSP, the configuration is basically the same. The one major difference is in the RADIUS configuration. Because a VSP can already have a RADIUS server configured, you can simply reference that server from the VSP-level. When setting up RADIUS-based access from the top-level, you must also configure the server properties.

Directories

Each access point includes a set of user directories and a set of permission definitions. The user directory contains an authentication database of locally configured users. In addition, you can configure other authentication directory types, such as RADIUS.

Note: The order in which you configure the directories establishes the order in which the AA-SBC checks directories for authentication. For example, if you want to override a users privileges as they are set in the RADIUS directory, configure the static users directory first.

If authentication succeeds, the permissions associated with that user are applied to all the subsequent operations.

Access object summary

The following table lists and briefly describes the **access** objects. See the following chapter for other objects in the CLI hierarchy:

- Chapter 75, "Virtual system partition (VSP) objects"

Object name	Description
`access`	Opens the access configuration object for editing.
`permissions`	Configures permission settings to apply to users and RADIUS servers.
`users`	Adds access for statically configured users.
`password-policy`	Specifies the password requirements for locally configured users.
`user`	Adds individual users to the system authentication directory.
`radius`	Configures AA-SBC to use a RADIUS server to perform user authentication.
`group`	Configures a RADIUS group for user authentication done outside of a VSP.
`server`	Configures the server associated with the RADIUS group.
`call-field-filter`	Configures the fields of the call detail record sent to the RADIUS server. See the call-field-filter object in the accounting for a description.
`enterprise`	Configures access for users from an enterprise directory group.

access

Purpose

Opens the access configuration object for editing. You can set access privileges from two points in the hierarchy, either:

- system-wide, from the top level of the CLI hierarchy
- per-vsp, from within each VSP object.

Syntax

```
config access
config vsp access
```

Properties

None

Example

```
NNOS-E> config access
config access>

NNOS-E> config vsp
config vsp> config access
config access>
```

permissions

Purpose

Opens or creates a set of permissions. From this object you can set access to a variety of box-level services.When a user successfully logs in, AA-SBC applies the permissions associated with that user to all subsequent operations.

Note that enabling permissions is not the same as enabling the service. To enable services, see the following chapters:

Property	Chapter reference
CLI	Chapter 13, “CLI objects”
AA-SBC Management System	Chapter 79, “Web objects”
User portal	*Net-Net OS-E – Using the NNOS-E Management Tools*
Config	Throughout this manual
Status	Chapter 4, “Status provider show commands”
Actions	Chapter 3, “Actions”
Call logs	Chapter 6, “Accounting objects” and Chapter 39, “Master services objects”
Templates	Chapter 80, “Web-service objects”
Web services	Chapter 80, “Web-service objects”
Debug	N/A

Enter a previously configured permissions set name to edit it or enter a new text string to create the permission set.

Syntax

```
config access permissions name
config vsp access permissions name
```

Properties

Property name	Description
cli {advanced | normal | disabled}	Sets access to the CLI. Select one of the following: • **advanced**—allows full access to CLI commands. • **normal**—allows partial CLI access. When restricted, users do not have access to advanced functionality such as debug tools, shell, etc. Further access is dependent on the other properties set in this object (**config**, **status**, **actions**). • **disabled**—prohibits access to the CLI. **Example:** set cli advanced The default setting is **normal**.
cms {enabled | enabled-web-only | disabled}	Sets access to the AA-SBC Management System. Select one of the following: • **enabled**—allows access to the AA-SBC Management System • **enabled-web-only**—allows access to the AA-SBC Management System only. • **disabled**—prohibits access to the AA-SBC Management System. **Example:** set cms enabled-web-only The default setting is **enabled**.

Property name	Description
user-portal {enabled | enabled-advanced| disabled}	Sets the user access to the portal feature of the AA-SBC Management System. See *Net-Net OS-E – Using the NNOS-E Management Tools* for complete information on this feature. Select either: • **enabled**—sets the user portal to display call and IM data. When **enabled**, and all other permission properties are disabled, the user is taken directly to the portal page when logging into the AA-SBC Management System. If other properties are enabled as well, the user is taken to the AA-SBC Management System home page, and the portal tab is available for selection. • **enabled-advanced**—sets the user portal to display session data, in addition to the standard call and IM data. • **disabled**—disables the user portal. **Example:** set user-portal enabled The default setting is **disabled**.
config {enabled | view | disabled}	Sets access to system configuration commands. These commands are used to change the running configuration. Select either: • **enable**—allows full access to config commands. • **view**—allows users to view the system configuration, but prevents them from executing config commands. • **disabled**—prohibits access to config commands. **Example:** set config view The default setting is **enabled**.
status {enabled | disabled}	Enables (allows) or disables (prohibits) the ability to execute system **show status** commands. These commands display various components of system status and data. **Example:** set status enabled The default setting is **enabled**.

Property name	Description
actions {enabled | disabled}	Enables (allows) or disables (prohibits) the ability to execute system actions. An action is a command that immediately acts on AA-SBC and one of its components. **Example:** set actions disabled The default setting is **enabled**.
call-logs {enabled | disabled}	Enables (allows) or disables (prohibits) access to the system accounting functions and call-log data. Accounting functions include RADIUS and Diameter accounting services, system logging (syslog), the accounting database, and the accounting file system. Call logs include user-specific session, whole session, and SIP message logs. **Example:** set call-logs disabled The default setting is **enabled**.
templates {enabled | disabled}	Sets the ability to use the web services template API. Templates provide access to a bundled configuration process that simplifies the use of web services by automating aspects of the configuration. For example, you could create a template to automate provisioning of AA-SBC devices. When **enabled**, the user can access the template interface; when **disabled**, the user cannot. You must enable **web-services** permissions for access to the template API. Additionally, this permission provides read-only access. You must also enable other permissions (e.g., config, status, and actions) for full web services capabilities. **Example:** set templates disabled The default setting is **enabled**.

Property name	Description
troubleshooting {enabled | disabled}	Sets the ability to use the troubleshooting web service. AA-SBC provides a troubleshooting web service that accesses the call database and sends troubleshooting requests to the AA-SBC device for call binding information. When **enabled**, the user can access the troubleshooting web service; when **disabled**, the user cannot. **Example:** set troubleshooting disabled The default setting is **enabled**.
web-services {enabled | disabled}	Sets the ability to initiate WSDL requests through the web services management API. When **enabled**, the user can access the web service interface; when **disabled**, the user cannot. Note that this permission provides read-only access. You must also enable other permissions (e.g., config, status, and actions) for full web services capabilities. Enable template permissions for access to the **template** API. **Example:** set web-services disabled The default setting is **enabled**.
debug {enabled | disabled}	Enables (allows) or disables (prohibits) the ability to access debug commands. When enabled, the user has shell and debug access; when disabled, the user does not. Typically, these commands, which are a licensed feature, are not for end-user use. If not licensed, the **debug** property does not display. **Example:** set debug disabled The default setting is **disabled**.
login-attempts {*integer* | unlimited}	Specifies the maximum number of failed login attempts allowed by the AA-SBC device. When this value is reached the user is locked out until an administrator either configures a new password or executes the “login unlock” action for that user. **Example:** set login-attempts 3 If setting a number, enter a value between 3 and 12. The default setting is **unlimited**.

Property name	Description
permitted-view	Assign a permitted view you want a user to have. If no permitted-view is specified, the default permitted view is set to **all**. The following are valid permitted views: • all • minimal • basic • secureAccessProxy • secureMediaProxy • lcs • sametime • imFederation • e911 • phoneServices • pstn • csta • security-admin • security-operator • sip-admin **Example**: **set permissions security-admin** The default setting is **all**.
lcr-import	Reserved for future use.

users

Purpose

Opens the users directory for configuration. When setting up authentication through this directory, you statically add users, and their privileges, to the system authentication database. Alternatively, you can configure AA-SBC to perform authentication via a RADIUS server with the users object.

Syntax

```
config access users
config vsp access users
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables access for statically configured users. **Example:** set admin enabled The default administrative state is **enabled**.

password-policy

Purpose

Specifies the password requirements for locally configured users. It is through this object that you define string requirements, reusability, and expiration times.

Syntax

```
config access users password-policy
config vsp access users password-policy
```

Properties

Property name	Description
duration {unlimited | *days*}	Specifies the length of time, in days, for which a password is valid. Enter either a number between 1 and 365, or select unlimited, which indicates that the password never expires. When a password expires, AA-SBC prompts you to change it on your next log in and sends a message to the event log. **Example:** set duration 7 The default number of days is **unlimited**.
minimum-length *characters*	Specifies the minimum number of characters allowed for a password. **Example:** set minimum-length 5 Enter a value between 2 and 64. The default minimum length is **4** characters.
character-types *integer*	Specifies the number of different character types allowed in a password. The character type choices are uppercase, lowercase, numeric, and other (anything non-alphanumeric). **Example:** set character-types 3 Enter a value between 1 and 4. The default number of types is 1.

Property name	Description
allow-sequences {true | false}	Specifies whether the password can contain sequences or repeated characters. If set to **true**, any string is acceptable (if it meets the other property constraints). If set to **false** you cannot include a sequence or repeated character in a password. A sequence is considered two or more consecutive numbers or letters (ab, 67, or MN, for example). Characters are considered repeated only if they are directly next to each other (skiing would be invalid, banana would be allowed). **Example:** set allow-sequences false The default setting is **true**.
recycle-check {*integer* | disabled}	Specifies whether and when a password can be reused. If set to **disabled**, any password can be reused. Specifying a number indicates the number of previous passwords that cannot be reused. For example, specifying four requires that a new password not be the same as any of the last four passwords. **Example:** set duration enabled The default setting is **disabled**.

user

Purpose

Adds the named user to the system authentication database and assigns a previously configured set of permissions.

Enter a user name for the entry; enclose the name in quotation marks if it contains spaces.

Syntax

```
config access users user name
config vsp access users user name
```

Properties

Property name	Description
password *string*	Configures a password for the named user. A password string must be at least four characters long. **Example:** `set password admin` `confirm: ***************` There is no default setting.
permissions *reference*	Associates a set of permissions with the named user. These permissions include access to a variety of box-level services. See the permissions object for details. Enter a previously configured set of permissions. **Example:** set permissions vsp access permissions grantAll There is no default setting.

radius

Purpose

Configures AA-SBC to use a RADIUS server to perform user authentication and sets basic RADIUS functionality. For system-wide access use the group and server objects to define and identify the server. For VSP access, use the **group** property within this object. Alternatively, you can statically configure users for authentication and privileges via the users object.

Note: The radius subobject is applicable to the access object whether you configure it from the top level of the CLI hierarchy or from within a VSP. However, the group property, which references a previously configured RADIUS group, is only available from within a VSP. When configuring RADIUS from outside of the VSP, you must create a new group and server.

Syntax

```
config access radius
config vsp access radius
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables the RADIUS server authentication configuration. When enabled, the AA-SBC device forwards authentication requests to the specified RADIUS server. **Example:** set admin enabled The default administrative state is **enabled**.
`group` *`groupReference`* **VSP access only**	Specifies the RADIUS group that AA-SBC uses for user authentication. A RADIUS group defines the authentication and accounting services associated with a group of RADIUS servers, configured using the VSP radius-group object. Enter a reference to a previously configured group. **Example:** set group “vsp radius-group mgmtEmployees” There is no default setting.

Property name	Description
`default-permissions` *`reference`*	Associates a set of permissions to apply if there are no specifically configured permissions in place. These permissions include access to a variety of box-level services. See the permissions object for details. Enter a previously configured set of permissions. **Example:** set default-permissions vsp access permissions grantAll There is no default setting.

Property name	Description
`default-sip-address` *`regExp replacement`*	Specifies the SIP address to use when displaying calls via the portal. When the portal is configured for a user, they only see their own calls in the AA-SBC Management System. In order to filter for the user, AA-SBC needs to know the SIP address. This can be set on the RADIUS server. If there is not a SIP address defined for the user in the RADIUS server, AA-SBC uses this property to generate a SIP address from the access user name. • **regExp**—enter a regular expression identifying the portion of the attribute to match. For example, the following expression identifies a subexpression (between the parenthesis) that matches all names: (.*) • **replacement**—enter a string that defines how to recompose the resulting regExp string. The replacement string is what AA-SBC searches on when displaying calls in the portal for that user. In the following example, the first component from the regular expression is substituted in place of the "1" and appended to "@company.com." \1@company.com **Example:** set default-sip-address (.*) \1@company.com There is no default setting. For more information regarding configuring regular expressions and replacement strings, see the section in Chapter 1, "Using regular expressions" on page 36.

group

Purpose

Configures a RADIUS group allowing the AA-SBC device (the RADIUS client) to perform user authentication for user access. (To setup authentication of SIP traffic, use the VSP radius-group object.) Associate servers with the group using the server object.

This object is only available when configuring user access outside of the VSP. Specify the RADIUS group name using up to 16 alphanumeric characters with no blank spaces.

Syntax

```
config access radius group name
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables the RADIUS authentication and accounting server configuration. When enabled, authentication and SIP call accounting records are forwarded to the specified server IP address and port numbers. **Example:** set admin enabled The default administrative state is **enabled**.
accounting-mode {duplicate | round-robin | fail-over *retryNumber*}	Sets the RADIUS group accounting operational algorithm: round-robin, failover, or duplicate. • **round-robin**—If you configure multiple accounting servers in the accounting group, the round robin algorithm performs continued accounting requests to primary and secondary servers until a valid accounting response is received. • **duplicate**—The duplicate algorithm issues multiple duplicate accounting requests to all servers in the RADIUS accounting group. A duplicate accounting request uses the same client source IP address and source UDP port. • **fail-over**—If you configure multiple accounting servers, the failover algorithm forwards accounting requests to secondary servers should the current accounting server fail. You can specify up to 256 failover servers. **Example:** `set accounting-mode round-robin` The default setting is **duplicate**.

Property name	Description
authentication-mode {round-robin| fail-over *retryNumber*}	Sets the RADIUS group authentication operational algorithm: round-robin or failover. • **round-robin**—If you configure multiple authentication servers in the RADIUS group, the round robin algorithm performs continued authentication requests to primary and secondary servers until a valid authentication response is received. • **fail-over**—If you configure multiple authentication servers in the RADIUS group, the failover algorithm forwards authentication requests to secondary servers should the current authentication server fail. You can specify up to 256 failover attempts to other servers. **Example:** `set authentication-mode round-robin` The default setting **fail-over 3**.
type *recordType*	Sets the type of SIP accounting record to use. Currently, the only valid SIP accounting record type is Cisco. **Example:** `set type cisco` The default setting is **cisco**.

Property name	Description
included-in-default {true | false}	Specifies if this RADIUS group is to be included in the default RADIUS authentication and accounting target group. If set to **true**, authentication and accounting requests are forwarded to this group if there are no configured policies that govern or redirect RADIUS requests to other servers. **Example:** `set included-in-default true` The default setting is **true**.
send-digest-contents {true | false}	Specifies whether to include the SDP contents in the RADIUS Auth-Request message. If set to **true**, AA-SBC does include the contents. Note that this feature is for customized RADIUS use. If you enable it for a RADIUS server that does not support this option, the RADIUS server will then reject every RADIUS request. **Example:** `set` send-digest-contents `true` The default setting is **false**.

server

Purpose

Identifies and defines the operating parameters of the RADIUS server(s) for a specified group. This object is only available when configuring user access outside of the VSP.

Enter the host name or IP address for your RADIUS server.

Syntax

```
config access radius group name server host
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables the RADIUS authentication and accounting server configuration. When enabled, authentication and SIP call accounting records are forwarded to the specified server IP address and port numbers. **Example:** set admin enabled The default administrative state is **enabled**.
authentication-port *portNumber*	Sets the UDP port number that the RADIUS client (AA-SBC device) uses to send authentication requests to the RADIUS server. **Example:** `set authentication-port 998` Enter a value between 1 and 65535. The default setting is port **1812**.
accounting-port *portNumber*	Sets the UDP port number that the RADIUS client (AA-SBC device) uses to send accounting requests to the RADIUS server. **Example:** `set accounting-port 999` Enter a value between 1 and 65535. The default setting is port **1813**.
secret-tag *string*	Specifies the shared secret used to authenticate transactions between the AA-SBC device and the RADIUS server. The specified shared secret is never sent over the network. Note that the secret you enter here is a shared secret, and must match the secret configured on the RADIUS server. See Understanding passwords and tags for a description of the AA-SBC password handling. Enter up to 32 alphanumeric characters or less. **Example:** `set secret abc123xyz` There is no default setting.

Property name	Description
timeout *milliseconds*	Specifies the time (in milliseconds) to elapse before an accounting or authentication request to a RADIUS server times out. If the request times out, the request is retried for the specified number of attempts before the request is forwarded to the next RADIUS server in the configuration or dropped. **Example:** `set retries 5` Enter a value between 1 and 65535. The default setting is port **1000** milliseconds (1 second).
retries *`numberOfRetries`*	Sets the number of times AA-SBC retransmits an accounting or authentication request if the RADIUS server does not respond. **Example:** `set retries 5` Enter a number of retry attempts between 2 and 5. The default setting is **3** attempts.
window *`numberOfSessions`*	Configures the maximum number of simultaneous RADIUS client requests (authentication and accounting) sent to the RADIUS server. **Example:** set window 20 Enter a number of between 1 and 127. The default setting is **32**.

call-field-filter

Purpose

Configures the specific fields of the call detail record that AA-SBC should send to the target RADIUS server(s). See the accounting call-field-filter object description for complete details.

Syntax

```
config access radius group name call-field-filter
```

enterprise

Purpose

Applies access permissions to a group that is already defined in an enterprise directory server.This could be a group created in any number of ways—for example, as part of the directory setup and inherited by the AA-SBC device or through the directories group object.

You can configure permissions for any number of groups through this object, but can only map a group to one set of permissions.

Syntax

```
config access enterprise
config vsp access enterprise
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables the application of the specified permissions to the identified group. **Example:** set admin enabled The default setting is **enabled**.

Property name	Description
directory *directoryReference*	Specifies the directory server from which AA-SBC derives its user information. Enter a reference to a previously configured enterprise directory. **Example:** set server vsp enterprise directories active-directory employees There is no default setting.
group-mapping *groupReference permissionReference*	Maps previously defined directory groups to a set of previously configured permissions. Enter a group name that is recognized on the specified **directory**. Enter a reference to the permissions (configured using the permissions object), enclosing the reference in quotation marks. **Example:** set group-mapping marketing “vsp access permissions viewOnly” There is no default setting.

6. Accounting objects

Accounting description

The accounting object allow you to configure AA-SBC RADIUS and Diameter accounting services, system logging (syslog), the accounting database, and the accounting file-system. You can configure one or more of these accounting methods for capturing SIP call detail records. Note that if you want to include RTP statistics in your accounting records, you must enable the session-config media object **rtp-stats** property.

Accounting object summary

The following table lists and briefly describes the **accounting** objects. See the following chapter for other objects in the CLI hierarchy:

- Chapter 75, “Virtual system partition (VSP) objects”

Object name	Description
accounting	Opens the accounting configuration object.
radius	Enables and disables RADIUS accounting and references a configured server.
diameter	Enables and disables Diameter accounting and references a configured server.
database	Enables and disables the accounting database.
group	Configures the named database or syslog group configuration.
server (for database)	Configures the system to access the named database server.
call-field-filter	Specifies fields from the CDR to be written to the database target.
syslog	Enables and disables the system syslog server.

Object name	Description
group	Configures the syslog group configuration.
server (for syslog)	Configures the system to access the named syslog server.
call-field-filter	Specifies fields from the CDR to be written to the syslog server.
file-system	Enables and disables saving accounting records to a file-system on the system.
external-file-system	
path	Configures the file-system path object, which sets a target and operating characteristics.
call-field-filter	Specifies fields from the CDR to be written to the system file system.
archiving	Enables archiving of accounting and SIP call records off the system to a different server.
windows-share	Enables record archiving to a Windows server partition.
ftp-server	Enables record archiving to an FTP server.
http-server	Enables record archiving to an HTTP server.
smtp-server	Enables record archiving to an SMTP server.
db-server	Enables record archiving to a database server.
local	Enables record archiving to a location on the system.

accounting

Purpose

Configures RADIUS accounting, Diameter accounting, system logging, the accounting database, and the system log (syslog). See the table under the **call-field-filter** object for lists of the fields present in the call detail record. Also, see Saving accounting records for information on the accounting records file system.

Syntax

```
config vsp accounting
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables the accounting services associated with this VSP. Specifically, it controls forwarding of accounting and SIP call detail records to the included RADIUS or Diameter server, syslog or database server, or to the specified file-system. To control accounting services on the entire AA-SBC device, use the master-services object. (See Chapter 39, “Master services objects” for more information.) **Example:** `set admin enabled` The default setting is **enabled**.
duration-type {default | rounding | plus1}	Specifies the formula to use when calculating the duration of a call in milliseconds. Select either: • **default**—performs a standard disconnect-time minus connect-time/1000 calculation to determine actual seconds. • **rounding**—adjusts the call duration either up or down. If the disconnect-time minus connect-time/1000 calculation results in *n*+500 ms, the call duration rounds up to the next whole integer; if less than 500 ms, the call duration rounds down to the previous whole integer. • **plus1**—Performs a standard disconnect-time minus connect-time/1000 calculation and advances the value to next whole integer if not at zero. **Example:** `set duration-type rounding` The default setting is **default**.
retention-period *days*	Specifies the number of days that the system retains accounting records before purging them from the file system. Use the **purge-criteria** property to configure conditions for purging. **Example:** `set` retention-period `14` Enter a value between 0 and 21; the default setting is **7** days.

Property name	Description
subdirectory-size *records*	Specifies the number of records the system should write to each subdirectoy of the accounting root directory. The root directory is set with the services data-locations object. **Example:** `set` subdirectory-size `1500` Enter a value between 100 and 2000; the default setting is **1000** records.
purge-criteria {purge-always | purge-only-when-complete}	Specifies the criteria to use when deleting records from the file system. Select either: • **purge-always**—the system deletes all qualifying records when the retention period has expired, regardless of whether they were written to their intended targets. • **purge-only-when-complete**—the system only deletes those records that were successfully written to their defined targets. Any records not written are saved in the files system, even if they are expired, until the system can write them as configured. **Example:** `set` purge-criteria purge-only-when-complete The default setting is **purge-always**.
report *name acctField regExp category*	Configures the accounting service to generate summary reports which can then be viewed using the **show accounting-cdr-summary** status provider. The provider reports number and length of calls. This property defines how to categorize the data by specifying the CDR field from which the category is derived. The example below creates a report, Calling1, that categorizes call data records based on the SIP address in the From field. **Example:** `set report Calling1 From .*<sip:(.*)>.* "\1"` There is no default setting.

Property name	Description
purge-accounting-files {disabled | enabled}	Specifies whether to allow the internal purge function. When **enabled**, AA-SBC keeps files for the duration configured with the **retention-period** property. (Frequency of the purge function is not user configurable.) When **disabled**, the system does not purge files automatically. Instead, you must use the `accounting` **purge** action to remove accounting files from the system. This property is for accounting service performance tuning; do not change the value unless instructed to do so by Technical Support. **Example:** `set purge-accounting-file disabled` The default setting is **enabled.**
purge-check-interval *interval*	Specifies the frequency with which AA-SBC checks for accounting records that have exceeded the retention period. In the example below, AA-SBC checks for records once a day (or every 24 hours). Enter an interval in HH:MM:SS format. Optionally, you can enter the parameters surrounded by quotation marks, such as “1 days 12:00” for a 36-hour interval. **Example:** `set purge-check-interval` 24:00:00 The default setting is **0 days 12:00** (every 12 hours from the prior purge check).

Property name	Description
purge-disk-utilization-percent *percentage*	Specifies the percentage of disk utilization allowed before accounting records are purged from the disk.Setting this prevents the system disk from becoming overloaded with accounting records. Enter a value from 0 to 100 (percent). Entering 0 causes the purge process to start, causing the system to purge new records as they are written to disk. **Example:** `set purge-disk`-utilization-percent 50 The default setting is **90%**.
mirror-cdrs {yes | no}	When enabled, raw CDRs are mirrored among peers of a cluster. This permits failovers with no loss of CDRs. This is a secondary property and is also an advanced property. This setting should not be changed without being explicitly told to do so by Acme personnel. **Example**: **set mirror-cdrs no** The default setting is **yes**.

radius

Purpose

Opens the Remote Authentication Dial In User Service (RADIUS) configuration object on AA-SBC. Use the group property to reference a named RADIUS authentication and accounting group configuration that you created with the radius-group configuration object.

Syntax

```
config vsp accounting radius
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables RADIUS accounting. When **enabled**, the system forwards session archive records to the referenced RADIUS accounting group and server. **Example:** set admin enabled The default setting is **enabled**.
group *radiusGroupReference*	Specifies a previously configured RADIUS group, which allows the system (the RADIUS client) to perform user authentication and to forward accounting and SIP call detail records to the RADIUS servers. **Example:** set group vsp radius-group East There is no default setting.

diameter

Purpose

Enables and disables the Diameter configuration object on AA-SBC and sets the server(s) used to store records. The system, operating as a Diameter client, sends an accounting request to the Diameter server. The Diameter server returns an accounting response to the client indicating that it has received and processed the accounting request.

Use the group property to reference a named Diameter authentication and accounting group configuration that you created with the Diameter group configuration object.

Syntax

```
config vsp accounting diameter
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables RADIUS accounting. When **enabled**, the system forwards session archive records to the referenced DIAMETER accounting group and server. **Example:** set admin enabled The default setting is **enabled**.
group *diameterGroupReference*	Specifies a previously configured Diameter group, which allows the system (the Diameter client) to perform user authentication and to forward accounting and SIP call detail records to the Diameter servers. **Example:** set group vsp diameter group West There is no default setting.

database

Purpose

Enables and disables AA-SBC accounting database configuration object. Use the group subobject to set the characteristics of the database group, including queue and batch settings.

Syntax

```
config vsp accounting database
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables AA-SBC accounting database. When **enabled**, the system forwards session archive records to the specified database accounting group and server. **Example:** `set admin enabled` The default setting is **enabled**.

group

Purpose

Opens the specified database or syslog group and sets the group operational parameters. The properties available for the two objects differ, as noted in the descriptions.

Database batch properties

The database inserter for accounting call detail records (CDRs) collects a number of insert requests before trying to write them. (This effectively creates batch jobs.) You can configure AA-SBC to build these batches based on size or elapsed time by using the group **insert-batch-size** or **insert-batch-time** properties.

Syntax

```
config vsp accounting database group name
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables AA-SBC syslog or database accounting group configuration. When **enabled**, the system forwards session archive records to the specified group and server. **Example:** `set admin enabled` The default setting is **enabled**.
Database only mode {duplicate | fail-over}	Sets the database group operational algorithm. Select one: • **duplicate**—The duplicate algorithm issues multiple duplicate accounting requests to all servers in the accounting group. A duplicate accounting request uses the same client source IP address and source UDP or TCP port. • **fail-over**—If you configure multiple accounting servers, the failover algorithm forwards accounting requests to a secondary server should the current accounting session fail. If that server is unavailable, the system tries the next, and so on. **Example:** `set mode fail-over` The default setting is **duplicate**.

Property name	Description
Syslog only format {covergence | csv | tab | xml}	Sets the syslog file format to use when writing syslog records to servers included in this accounting group. Select a format: • **covergence**— text file format. • **csv**—Comma-separated values format. CSV format is a generic file format used for importing data into databases or spreadsheets, such as Microsoft Access or Excel (or several other database systems). CSV uses the .CSV file extension. • **tab**—Tabular format. • **xml**—eXtensible Markup Language format; for use with XML and Web applications. **Example:** `set format csv` The default setting is **covergence**.
Database only column-replacement-names *column alias*	Allows you to change the names of accounting record fields from the default to an alias. Use this, for example, to write to an external database with column names different than the AA-SBC internal column names. Enter an existing internal column name followed by the desired alias. **Example:** `set column-replacement-names From Caller` There are no default values.
batch-insert-size	The number of CDRs in one database insert request. EXAMPLE: **set batch-insert-size 40** Min: 1 / Max: 50 The default setting is **25**.

server (for database)

Purpose

Opens the specified database accounting server configuration. Sending data to the database server records the placement of the call; use the archiving and db-server objects to configure persistent storage of the contents of the call.

Accounting objects

Syntax

```
config vsp accounting database group name server name
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables the system accounting database server configuration. When **enabled**, the system forwards session archive records to the specified database group and named server. **Example:** `set admin enabled` The default setting is **enabled**.
type {local | oracle *`IPaddress`*:*`port`* [*sid*] | postgres *`IPaddress`*:*`port`* [*`databaseName`*] | sqlserver *`IPaddress`*:*`port`* [*`databaseName`*] | generic *`url driver validation`*}	Specifies the information necessary for the system to connect to the database server containing the accounting data (the destination to which the system forwards network traffic and SIP call accounting messages). For type **oracle**, **postgres**, or **sqlserver**, enter the database server type followed by the IP address and known TCP port number. Depending on the type of database server you are using, an optional information field is available for server IDs, names, etc. **Example:** **`set type postgres 192.168.43.8:543`**2 Contact your database server administrator for TCP port number assignments and any optional information. There is no default setting.

Property name	Description
username *name*	Sets the required username for accessing this database server. If the **type** property is set to **local**, the username must be **postgres**. Enter up to the maximum number of alphanumeric characters (with no blank spaces) allowed for the type of server you are using. If you are unsure, contact your database server administrator. **Example:** `set username administrator` There is no default setting.
password-tag *password*	Specifies the tag associated with the shared secret used to authenticate transactions between AA-SBC and this server. If the **type** property is set to **local**, the password-tag must point to a shared secret of **postgres**.See Understanding passwords and tags for information on the two-part password mechanism. **Example:** `set password-tag xyz123abc` There is no default setting.

call-field-filter

Purpose

Configures, for each group or path of a target type, the specific fields of the call detail record that AA-SBC should send to the target. The following table lists the fields available to the **fields** property. There is a subset of required minimum fields that are sent regardless of the configuration. Those fields are indicated with bold type. If you do not configure this object (set the **fields** property), all fields are sent to the target. See the *Net-Net OS-E – Session Services Configuration Guide*, Appendix B, for more information about call details records.

Field	Description
SessionId	**The unique internal identifier for the session.**
Recorded	An indicator as to whether the call was recorded or not.

Field	Description
CallId	**The unique call identifier from the user agent.**
To	**The To: URI.**
From	**The From: URI.**
Method	The SIP method that initiated the session.
IncomingRequestURI	The Request URI for the incoming call leg.
PreviousHopIp	The IP address of the previous hop in the call.
PreviousHopVia	The Via: header for the previous hop.
OutgoingRequestURI	The Request URI for the outgoing call leg.
NextHopIp	The IP address of the next hop.
NextHopDn	The domain name of the next hop.
Header	An arbitrary header from the call.
Origin	The origin header from the call.
SetupTime	**The time at which the call was set up.**
ConnectTime	**The time at which the call was connected.**
DisconnectTime	**The time at which the call was disconnected.**
DisconnectCause	The reason for disconnection.
Duration	**The duration of the call, in seconds.**
scpName	The VSP that handled the call.
CallID2	The secondary call identifier for the outgoing leg.
OrigGW	The name or endpoint string of the From server.
TermGW	The name or endpoint string of the To server.
PacketsReceivedOnSrcLeg	The total number of packets on the source leg that were successfully received.
PacketsLostOnSrcLeg	The total number of packets on the source leg that were lost.
PacketsDiscardedOnSrcLeg	The total number of packets associated with the call that were successfully discarded due to a UDP or TCP problem (e.g., an out-of-sequence error) on the source leg.
PdvOnSrcLeg	The average packet delay variation (jitter) experienced on the source leg.

Field	Description
MaxJitterOnSrcLeg	The maximum packet delay variation (jitter) experienced on the source leg.
CodecOnSrcLeg	The CODEC associated with the call on the source leg.
MimeTypeOnSrcLeg	The MIME type associated with the call on the source leg.
LatencyOnSrcLeg	The average amount of time spent processing packets on the source leg.
MaxLatencyOnSrcLeg	The maximum amount of time spent processing a packet on the source leg.
PacketsReceivedOnDestLeg	The total number of packets on the destination leg that were successfully received.
PacketsLostOnDestLeg	The total number of packets on the destination leg that were lost.
PacketsDiscardedOnDestLeg	The total number of packets associated with the call that were successfully discarded due to a UDP or TCP problem (e.g., an out-of-sequence error) on the destination leg.
PdvOnDestLeg	The average packet delay variation (jitter) experienced on the destination leg.
MaxJitterOnDestLeg	The maximum packet delay variation (jitter) experienced on the destination leg.
CodecOnDestLeg	The CODEC associated with the call on the destination leg.
MimeTypeOnDestLeg	The MIME type associated with the call on the destination leg.
LatencyOnDestLeg	The average amount of time spent processing packets on the destination leg.
MaxLatencyOnDestLeg	The maximum amount of time spent processing a packet on the destination leg.
Rx1000FactorOnDestLeg	The call quality attribute (voice quality score) used to calculate MOS scores on the destination leg.
Rx1000FactorOnSrcLeg	The call quality attribute (voice quality score) used to calculate MOS scores on the source leg.
MOSFmtOnDestLeg	The MOS score on the destination leg.
MOSFmtOnSrcLeg	The MOS score on the source leg.

Accounting objects

Field	Description
callType	The call type.
disconnectErrorType	The type of error that caused the disconnection.
ani	The caller ID associated with the automatic number identification.
callSourceRegid	Server name if available, or user portion of the FROM (source) or TO (destination) URI on the source leg.
callDestRegid	Server name if available, or user portion of the FROM (source) or TO (destination) URI on the destination leg.
newAni	The caller ID for the ANI after any manipulation is done by AA-SBC.
cdrType	The record type, either start or stop.
huntingAttempts	The number of times the system used the arbiter to select a dial plan and a failure occurred (including subsequent attempts).
callPDD	The post dial delay from the initial INVITE until the 200 OK.
callSourceRealmName	The call source domain name.
callDestRealmName	The call destination domain name.
callDestCRName	The name of the dial plan that forwarded the call.
in_peer_dst	The IP address and port of the destination phone to which the system forwarded the inbound call leg.
in_anchor_src	The IP address and port at the system platform where the inbound call leg was forwarded to the destination peer.
in_anchor_dst	The IP address and port at the system platform where the inbound call leg was received from the source peer.
in_peer_src	The IP address and port of the source phone that contacted the system over an inbound call leg.
out_peer_dst	The IP address and port of the destination phone to which the system forwarded the outbound (return) call leg.
out_anchor_src	The IP address and port at the AA-SBC device where the outbound call leg was forwarded back to the source peer.

Field	Description
out_anchor_dst	The IP address and port at the AA-SBC device where the outbound (responding) call leg was received from the destination peer.
out_peer_src	The IP address and port of the responding destination phone from which an outbound call leg was returned to AA-SBC.
calledPartyAfterSrcCallingPlan	The called party number after any manipulation on leg 1, but before any manipulation on leg 2.
lastStatusMessage	An integer indicating SIP message type last status message (omitting “200 OK”) and therefore call progress.
LastMediaPktTimestampOnDestLeg	The time of the last media packet on the destination leg.
LastMediaPktTimestampOnSrcLeg	The time of the last media packet on the source leg.
setupTimeInt	A millisecond representation of the setup-time field in Unix time. This is the number of milliseconds elapsed between midnight January 1, 1970 and the origination of the call.
incomingURIStripped	A version of the incoming-uri field with the extraneous information removed, leaving just the regular form of the URI.
dnis	**The Dialed Number Identification Service.**
newDnis	**The New Dialed Number Identification Service.**
customData	User-inserted fields.

Syntax

```
config vsp accounting database group name call-field-filter
config vsp accounting syslog group name call-field-filter
config vsp accounting file-system path name call-field-filter
config vsp radius-group name call-field-filter
```

Properties

Property name	Description
fields *fields*	Specifies the fields from the accounting record to send to the target. AA-SBC always inserts a set of minimum required fields. If you do not set this property, all fields from the CDR are sent to the target. If you do set this, the required minimum fields plus those you entered are sent. Enter multiple CDR fields separated by a plus sign (+) with no spaces. See the table under the **call-field-filter** object for a list of signaling-related CDR fields. **Example:** `set fields recorded+OrigGW+TermGW`

syslog

Purpose

Opens and enables AA-SBC syslog configuration object.

Syntax

```
config syslog
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables AA-SBC syslog. When **enabled**, the system forwards session archive records to the specified syslog group and server (for syslog). The system writes records in the log format specified in the group configuration. **Example:** `set admin enabled` The default setting is **enabled**.
port	Specifies the port number over which the AA-SBC should communicate with this syslog server. **Example**: set port 350 Min: 1 / Max: 65535 The default setting is **514**.

server (for syslog)

Purpose

Opens the AA-SBC syslog accounting server configuration using the specified IP address and UDP port number. The well-known UDP port number for syslog accounting is 514. Sending data to the syslog server records the placement of the call; use the archiving object to configure persistent storage of the contents of the call.

Syntax

```
config vsp accounting syslog group name server ipaddress:port
```

Properties

Property name	Description
name *text*	Specifies the syslog server name (or host name). Enter the name using up to 32 alphanumeric characters. **Example:** `set name syslogServer1` The default syslog server name is **CXC**.
admin {enabled | disabled}	Enables or disables AA-SBC syslog accounting server configuration. When **enabled**, the system forwards session archive records to the specified server IP address and port number. **Example:** `set admin enabled` The default setting is **enabled**.
facility {user | local0...local7}	Sets the user-defined syslog facility (**user** or **local0** to **local7**) to which AA-SBC logs accounting and SIP call detail records. Syslog facilities help isolate the origin of messages written to the syslog server. **Example:** `set facility local5` The default setting is **local0**.
priority {emergency | alert | critical | error | warning | notice | info | debug}	Sets the message priority to be associated with accounting and SIP call detail records. The system assigns all session archive records this priority before forwarding them to the syslog server. **Example:** `set priority warning` The default setting is **info**.
include-timestamp {true | false}	Appends a time stamp to each accounting record before forwarding the record to the syslog server. **Example:** `set include-timestamp true` The default setting is **true**.

file-system

Purpose

Opens and enables AA-SBC accounting file-system. This is the object through which you direct accounting and SIP call records to be saved to a target file path.

Syntax

```
config vsp accounting file-system
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables the AA-SBC file system. When **enabled**, the system forwards session archive records to the specified file system path. **Example:** `set admin enabled` The default setting is **enabled**.
format	The output format of the file you are creating. The following are valid file formats: -csv -proprietary -postgresql. **Example: set format postgresql** The default setting is **csv**.

Property name	Description
call-field-filter	Filter out what fields are sent with accounting records. If this is left blank, all fields are sent in the accounting records. The following are valid fields: -SessionID -Recorded -CallID -To -From -Method -IncomingRequestURI -PreviousHopIp -PreviousHopVia -OutgoingRequestURI -NextHopIp -NextHopDn -Header -Origin **Example**: **set call-field-filter method, sessionid, to, from**
file-path	Enter the path and name of the file to write the records. **Example**: **set file-path /nnos_common/acct/test.csv**
roll-over	Set the schedule for creating new log files. The following values are valid: -never—never renew the file -minute—renew the file once a minute -hourly—renew the file once an hour -daily—renew the file once a day **Example**: **set roll-over hourly** The default setting is **daily**.

Property name	Description
purge-old-logs {true | false}	Allows you to remove files modified early than the retention period, excluding the current file. You can identify the current file using the status provider. **Example**: **set purge-old-logs true** The default setting is **false**.
retention-period	Set the number of days logs should be retained on the file system. **Example**: **set retention-period 2** Min: 0 / Max: 5184000 The default setting is **3** days.

external-file-system

Purpose

Configures the external file system, which allows you to write accounting records to an outside server.

Syntax

```
config vsp accounting external-file-system url
```

Properties

Property name	Description
admin [enabled | disabled]	When enabled, the AA-SBC forwards accounting and SIP call detail records to the target file path. **Example**: **set admin disabled** The default setting is **enabled**
format	The output format of the file you are creating. The following are available file formats: -csv -proprietary -tab -postgresql **Example**: **set format postgresql** The default setting is **csv**.
url	Enter the URL of the external target to which you are sending CDRs. **Example**: **set url ftp://url1:tomsmith#1@10.33.5.10:/acct/test/**
cdr-processing	Specify how the CDRs are collected. The following are the available processes: batch—Write the file with a specified number of CDRs collected before the file is sent. Min: 0 / Max: 4294967295 -roll-over—Write the file with a specified roll-over policy. These can be **never**, **hourly**, **daily** and **per-minute**. -interval—Write the file at a specified interval. Min: 60 seconds / Max: 1036800 seconds (12 days) **Example**: **set cdr-processing batch 15000** The default settings are **batch 20000**, **roll-over hourly**, **interval 0**.

path

Purpose

Specifies the path to the which AA-SBC writes accounting and SIP call detail records. In addition, the **path** object specifies format, time-stamp, and other configuration information. You can configure multiple path objects, each identifying a different target file. AA-SBC writes records to all enabled paths.

Specify a path name using up to 32 alphanumeric characters with no blank spaces. If the file does not exist, the system creates a file by that name.

Understanding accounting record roll-over

You can set the interval with which AA-SBC starts a new accounting record using the **roll-over** property. When you set the interval to either daily or hourly, the system handles the data as follows:

1. AA-SBC stops writing records to the current file (for example, /cxc_common/acctg.csv). This current file is set with the **file-path** property. Any pending CDRs are queued in memory.
2. AA-SBC renames the current file, including a time and/or date stamp in the file name. For example, if roll-over is set to daily, /cxc_common/acctg.csv becomes /cxc_common/acctg_2008-06-09.csv.
3. AA-SBC creates a new file with the name specified in the **file-path** property and begins writing files to it.

Syntax

```
config vsp accounting file-system path pathName
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables AA-SBC accounting file system path. When **enabled**, the system forwards accounting and SIP call detail records to the target **file-path**. **Example:** `set admin enabled` The default setting is **enabled**.
format {csv | tab | covergence}	Sets the file format to use when writing accounting and SIP call detail records to this file-system path. • **covergence**—text file format. • **csv**—Comma-separated values format. CSV format is a generic file format used for importing data into databases or spreadsheets, such as Microsoft Access or Excel (or several other database systems).CSV uses the .CSV file extension. • **tab**—Tabular format. **Example:** `set type tab` The default setting is **csv**.
file-path *pathString*	Sets the target file-system path for logging accounting and SIP call detail records. A file-path consists of a valid AA-SBC directory path and file name. **Example:** `set file-path /cxc/logging/logfile1` There is no default setting.

Property name	Description
roll-over {never | hourly | daily}	Specifies the interval at which AA-SBC starts a new file for accounting records. See Understanding accounting record roll-over for details on the roll-over process. Select either: • **never**—the system maintains the original time as it was first applied to the log file. The log file will continue to build under this time stamp. • **hourly**—the system begins a new file at the beginning of each hour. • **daily**—the system begins a new file at the beginning of each day (at midnight, 00:00). **Example:** `set roll-over daily` The default setting is **never**.
purge-old-logs {true | false}	Specifies whether AA-SBC should age out and delete files from the system. When set to **true**, AA-SBC deletes rollover files that have expired according to the time frame set with the **retention-period** property. Purging is only applicable if the **roll-over** property is set to **hourly** or **daily**. **Example:** `set` purge-old-logs `true` The default setting is **false**.
retention-period *days*	Specifies the number of days that rollover files should be kept on the system. **Example:** `set` retention-period `5` Enter a value from 1 to 21; the default setting is **3** days.

archiving

Purpose

Enables archiving of AA-SBC accounting and SIP call records off the system to a different server. Archiving is the persistent storage of the contents of the call (as opposed to the database or syslog server, which just records the placement of the call). Archiving can be initiated as an action (archive), as a scheduled task, or automatic through the **continuous-archiving** property.

You can specify the types of information to store with the *include-* properties. If you do not include any of the message types, the archive will contain just the meta data (To, From, setup/connect/disconnect times, and call ID). All message types are included by default.

When archiving, AA-SBC creates both a .zip file and an XML file of the archive contents. The XML file contains all of the XML data for the call except for the SIP messages. The .zip file contains the XML file and an additional file called sip.xml, which contains the SIP messages.

You must configure a server in one of the archiving subobjects for the archiving mechanism to work:

- windows-share
- ftp-server
- smtp-server
- db-server
- local

Syntax

```
config vsp accounting archiving
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables system archiving. When **enabled**, the system forwards session archive records to the specified server or database. **Example:** `set admin enabled` The default setting is **enabled**.
archive-timeout *minutes*	Specifies the number of minutes a status query can run before causing the archiving process to fail. When the timer expires, the system cancels the archive request. Failure notification is written to the accounting log. This parameter does not time out an entire archive if the time to archive one *record* exceeds this threshold. Instead, it specifies the maximum time that an underlying *status query* can take to generate an archive record. (The system makes status queries when adding file transfers, raw media, and mixed media to an archive.) If a status query times out, the system fails the entire archiving operation. Because status query results may be quite large (and time-consuming to generate), this prevents adding records that would ultimately fail the archiving operation. **Example:** `set` archive-timeout `3` The default setting is **2** minutes.
continuous-archiving {true | false)	Specifies whether the archiving process runs automatically. When true, the process executes without need of scheduling or executing the archive action. When false, you must initiate the process manually. **Example:** set continuous-archiving true The default setting is **false**.

Property name	Description
use-compression {true | false)	Specifies whether the ZIP archive files should be compressed. If **true**, the system compresses the files, resulting in smaller archives. If **false**, the archiving process runs more quickly. **Example:** set use-compression false The default setting is **true**.
initial-archive-span *days*	Sets the number of days to of records archive prior to the day that archiving was first enabled. The system will then archive all call and IM records for that number of days into a single file. **Example:** set initial-archive-span 1 Enter a value from 1 to 30. The default setting is **7** days.
include-related-sessions {true | false}	Sets whether to merge sessions with the same call ID into a single archive. Use this in instances of an IM session with more than two participants or when an LCS tunneled client chats with an LCS or federated Sametime client. If set to **true**, system merges all related messages into a single archive and removes any duplicate messages (created by a multi-client chat) from that archive. If set to **false**, the system leaves multiple dialog archives in separate files. **Example:** `set` include-related-sessions `true` The default setting is **false**.
include-sip-messages {true | false}	Specifies whether to include the entire content of the SIP message, encoded in XML, in the archive record. **Example:** `set` include-sip-messages `false` The default setting is **true**.
include-file-transfers {true | false}	Specifies whether to include file transfer data in the archive record. When set to **true**, the system includes any files that were transferred during an IM session in the archive. **Example:** `set` include-file-transfers `false` The default setting is **true**.

Property name	Description
include-mixed-media {true | false}	Specifies whether to include WAV files resulting from recorded phone conversations in the archive record. Note that if this is set to true, and you have recorded a call that contains an unsupported CODEC, the archiving operation will fail. See recording-policy for more information. **Example:** `set` include-mixed-media `false` The default setting is **true**.
include-raw-media {true | false}	Specifies whether to include the media-related XML files in the archive record. These files contain the RTP payload and timing information. **Example:** `set` include-raw-media `false` The default setting is **true**.
include-instant-messages {true | false}	Specifies whether to include instant message traffic in the archive record. **Example:** `set` include-instant-message `false` The default setting is **true**.
include-urls {true | false}	Specifies whether to include URLs (captured in instant messages) in the accompanying XML file. If set to false, the URLs are not included in the XML file, but are included in the archive record. **Example:** `set` include-urls `false` The default setting is **true**.
include-audit-events {true | false}	Specifies whether to include events having to do with auditing in the archive record. **Example:** `set` include-audit-events `true` The default setting is **false**.

windows-share

Purpose

Enables archiving of AA-SBC accounting and SIP call records to a selected Windows server partition. You must also enable archiving through the archiving object.

Syntax

```
config vsp accounting archiving windows-share name
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables the AA-SBC Windows server partition used for archiving. When **enabled**, the system forwards session archive records to the specified partition. When **disabled**, the system does not write to the partition. **Example:** `set admin enabled` The default setting is **enabled**.
domain *domainName*	Specifies the partition to which the system writes archive records. Enter the Windows Domain name (which is not necessarily the DNS domain). **Example:** set domain ABCcompany There is no default setting.
username *name*	Specifies the required username for accessing this Windows server partition. This name must match the name configured on the Windows server, and must have write permission. **Example:** `set username administrator` There is no default setting.
password-tag *password*	Specifies the tag associated with the shared secret used to authenticate transactions between AA-SBC and this server. See Understanding passwords and tags for information on the AA-SBC two-part password mechanism. **Example:** `set password xyz123abc` There is no default setting.
server *ipAddress*	Specifies the IP address of the server that hosts this Windows server partition. **Example:** `set server 192.168.10.10` The default setting is **0.0.0.0** (no setting).

Property name	Description
port *portNumber*	Specifies the Server Message Block (SMB) port number over which the system should communicate with this Windows server partition. (This is the TCP port where SMB is hosted on the server.) **Example:** `set port 1776` Enter a value from 1 to 65535; the default port is **445**.
path *pathName*	Specifies the path to which the system should write the accounting archive records. **Example:** `set` path /accounting/archives/records1 There is no default setting.

ftp-server

Purpose

Enables archiving of AA-SBC accounting and SIP call records to a selected FTP server. You must also enable archiving through the archiving object.

Syntax

```
config vsp accounting archiving ftp-server name
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables the FTP server archiving configuration. When **enabled**, the system forwards session archive records to the specified FTP server. When disabled, the system does not write to the server. **Example:** `set admin enabled` The default setting is **enabled**.
username *name*	Specifies the required username for accessing this FTP server. This name must match the name configured on the FTP server. **Example:** `set username administrator` There is no default setting.
password-tag *password*	Specifies the tag associated with the shared secret used to authenticate transactions between AA-SBC and this server. See Understanding passwords and tags for information on the AA-SBC two-part password mechanism. **Example:** `set password xyz123abc` There is no default setting.
directory *directoryName*	Specifies the directory on the FTP server into which the system writes the archive records. **Example:** `set` directory /archives There is no default setting.
server *ipAddress*	Specifies the IP address of the system that hosts this FTP server. **Example:** `set server 192.168.10.10` The default setting is **0.0.0.0** (no setting).

Property name	Description
port *portNumber*	Specifies the port number over which the system should communicate with this FTP server. **Example:** `set port 1776` Enter a value from 1 to 65535; the default port is **21**.
timeout *milliseconds*	Specifies the time, in milliseconds, that the AA-SBC connection to the FTP server can remain open without a response before the system closes the connection. **Example:** `set` timeout 100000 The default setting is **60,000** milliseconds.

http-server

Purpose

Enables archiving of AA-SBC accounting and SIP call records to a selected HTTP server. You must also enable archiving through the archiving object.

Syntax

```
config vsp accounting archiving http-server name
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables the HTTP server archiving configuration. When **enabled**, the system forwards session archive records to the specified server. When **disabled**, the system does not write to the server. **Example:** `set admin enabled` The default setting is **enabled**.
directory *directoryName*	Specifies the directory on the HTTP server into which the system writes the archive records. **Example:** `set` directory /archives There is no default setting.
url *urlAddress*	Identifies, by means of a URL, the computer that hosts this HTTP server. **Example:** `set url www.companyABC.com` There is no default setting.
timeout *milliseconds*	Specifies the time, in milliseconds, that the AA-SBC connection to the HTTP server can remain open without a response before the system closes the connection. **Example:** `set` timeout 30,000 The default setting is **60,000** milliseconds.

smtp-server

Purpose

Enables archiving of AA-SBC accounting and SIP call records to a selected Simple Mail Transfer Protocol (SMTP) server. When enabled, AA-SBC sends out the archives in the form of an email attachment to the specified destination mailbox. You must also enable archiving through the archiving object.

Syntax

```
config vsp accounting archiving smtp-server name
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables the SMTP server archiving configuration. When **enabled**, the system emails session archive records to the specified address. When **disabled**, the system does not email archives. **Example:** `set admin enabled` The default setting is **enabled**.
server *host*	Specifies the IP address or host name of the outgoing mail server that handles these session archives. **Example:** `set server mail.companyABC.com` There is no default setting.
destination-mailbox *address*	Specifies the email address to which the system sends the session archives. **Example:** set destination-mailbox admin@companyABC.com There is no default setting.
reply-mailbox *address*	Specifies the address that appears in the “From” field of the emailed archives. If you do not specify this property, the “From” field is empty. **Example:** set reply-mailbox archives@companyABC.com There is no default setting.

Property name	Description
password-tag *password*	Specifies the tag associated with the shared secret used to authenticate transactions between AA-SBC and this SMTP server. See Understanding passwords and tags for information on the AA-SBC two-part password mechanism. **Example:** `set password xyz123abc` There is no default setting.
port *portNumber*	Specifies the port number over which the system should communicate with this SMTP server. **Example:** `set port 100` Enter a value from 1 to 65535; the default port is **25**.

db-server

Purpose

Enables archiving of AA-SBC accounting and SIP call records to a selected database server. You must also enable archiving through the archiving object. Note that this configuration defines the database archiving location. Use the server (for database) object to configure storage of accounting records.

Syntax

```
config vsp accounting archiving db-server name
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables the database server archiving configuration. When **enabled**, the system forwards session archive records to the specified server. When disabled, the system does not write to the server. **Example:** `set admin enabled` The default setting is **enabled**.
username *name*	Specifies the required username for accessing this database server. This name must match the name configured on the database server. **Example:** `set username administrator` There is no default setting.
password-tag *password*	Specifies the tag associated with the shared secret used to authenticate transactions between AA-SBC and this server. See Understanding passwords and tags for information on the two-part password mechanism. **Example:** `set password xyz123abc` There is no default setting.

Property name	Description
server *ipAddress*	Specifies the IP address of the system that hosts this database server. **Example:** `set server 192.168.10.10` The default setting is **0.0.0.0** (no setting).
url *urlAddress*	Identifies, by means of a Java Database Connectivity (JDBC) URL, the computer that hosts this database server. JDBC is s a standard for connecting to databases from Java. If you use a custom driver, you must specify the connection URL. Consult *www.java.sun.com* for the appropriate JDBC URL. **Example:** `set url www.companyABC.com` There is no default setting.
driver-class *class*	Specifies the driver class that implements the JDBC driver interface. (See **url**, above, for more information.) **Example:** set driver-class com.oracle.jdbc.Driver There is no default setting.

local

Purpose

Enables and specifies the directory and path locations on AA-SBC for saved accounting and SIP call records for this VSP. You must also enable archiving through the archiving object.

The **local** object works in conjunction with several other pieces of the system software:

- You can use the archive **vsp** action to initiate the backup immediately.
- You can use the task object to schedule automated backups.
- You set the number of days worth of records saved when maintenance is run using the VSP database object.
- You configure maintenance using the master services database object.

The **local** object allows archiving of call logs and associated recordings to the local hard drive. Use this in cases where a security model prohibits pushing files from network devices to storage devices. This allows you to instead “pull” the data off of AA-SBC to retrieve it.

Note: You must ensure that the archiving action is done more frequently than the shortest database history parameter (set in the VSP database object). This is because the database maintenance operation sweeps call log information that is older than the stored configured history. For example, if the call-detail-history is set to 3 days, database maintenance will sweep all call-detail records older than 3 days. Therefore, you must configure archiving to occur more often than once every 3 days.

Syntax

```
config vsp accounting archiving local name
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables the archiving at the local level. When **enabled**, the system forwards session archive records to the specified directory. When **disabled**, the system does not write to the directory. **Example:** `set admin enabled` The default setting is **enabled**.
directory *path*	Sets the location to which the system archives the current VSP, moving anything stored for a session off of the box. This includes recorded media, IM logs, file transfers, etc. Archiving frees up storage space; you can later reopen archives to view session information. **Example:** `set directory` /cxc/admin/archives The default path is **/cxc_common/archive**.

7. Admission Control objects

Admission control description

The **admission-control** object sets thresholds which determine the rate at which traffic is accepted onto the box. By controlling this level, you protect the box from dedicating too much processing power to, for example, call or registration processing. A traffic threshold level keeps the box stable, safeguards against attacks, and prevents AA-SBC from dropping calls because the CPU was overburdened.

Admission control on the VSP sets a rate limit on calls coming from outside of the network and through AA-SBC (before being forwarded to an upstream server). Note that you can also set admission and emission control on a server-pool-admission-control within a server pool. For a server, the emission settings control the rate at which AA-SBC can forward calls to the upstream server. Admission control on a server determines the rate that the upstream server can send calls back to AA-SBC.

Admission control object summary

The table below lists and briefly describes the **admission-control** object. See the following chapter for other objects in the CLI hierarchy:

- Chapter 75, "Virtual system partition (VSP) objects"

Object name	Description
`admission-control`	Sets the properties that control the allowable rate of calls in and out of the system.

admission-control

Purpose

Sets the thresholds used to calculate the admission control limit for the VSP. These values are used to derive a rate limit for calls coming from outside of the network and into AA-SBC.

Syntax

```
config vsp admission-control
```

Properties

Property name	Description
cpu-monitor-span *seconds*	Sets the number of seconds over which AA-SBC calculates the total system CPU average. At the conclusion of the span, the average value is compared to the call and registration CPU thresholds to determine whether to modify the dynamic threshold. The longer the span, the fewer the changes to the thresholds. A shorter span will result in reaction to brief CPU activity spikes. **Example: set cpu-monitor-span** Enter a value from 1 to 100. The default setting is **20** seconds.
cpu-monitor-interval *seconds*	Sets the frequency, in seconds, with which AA-SBC calculates the total system CPU average for the last span. **Example: set cpu-monitor-span** Enter a value from 1 to 100. The default setting is **10** seconds.

Property name	Description
registration-admission-control {enabled | disabled}	Enables or disables registration admission control on the VSP. If enabled, the controls set with the pending and dead registration high- and low-watermarks are applicable. This admission control suppresses new registrations to allow resolving registrations in progress, preventing “rate of registration” attacks. **Example:** set registration-admission-control enabled The default setting is **disabled**.
max-number-of-registrations *registrations*	Sets the maximum number of registrations allowed in the location cache. When the system reaches the maximum registration count, registrations are denied until the number falls below this threshold. Entries are cleaned out of the location cache at an interval specified by the **max-expiration** property set in the registration-service object. **Example:** set max-number-of-registrations 50000 Enter a value from 1 to 1,000,000. The default setting is **30,000**.
registrations-high-cpu-threshold *percent*	Sets an upper threshold, as a percentage, for registration processing average CPU usage. The registration dynamic threshold is calculated based on the admission-control/ pending-registrations-high-watermark property. When the average CPU usage exceeds this high threshold, AA-SBC decrements the dynamic threshold until it reaches the value set with the pending-registrations-low-watermark property or processing drops below the registrations-high-cpu-threshold value. **Example:** set registrations-high-cpu-threshold 75 Enter a value from 1 to 100. The default setting is **90** percent.

Property name	Description
registrations-low-cpu-threshold *percent*	Sets the low-end threshold, as a percentage, for registration processing average CPU usage, based on the registration dynamic threshold. When the SIP process CPU falls below the low threshold, AA-SBC increments the dynamic threshold by 20% of the pending-registrations-high-watermark value until it reaches the pending-registrations-low-watermark value. **Example:** set registrations-low-cpu-threshold 50 Enter a value from 1 to 100. The default setting is **70** percent.
pending-registrations-high-watermark *registrations*	Sets a hard limit for the number of in-progress registrations allowed before the system suppresses all registrations. Suppression stops if the in-progress registrations exceed the watermark. This is an absolute value for suppressing registrations. The system also suppresses registrations when CPU usage is above the dynamic threshold. **Example:** set pending-registrations-high-watermark 4500 Enter a value from 1 to 1,000,000. The default setting is **3000** registrations.

Property name	Description
pending-registrations-low-watermark *registrations*	Sets a hard limit for the number of in-progress registrations allowed before the system begins registration suppression. When below this threshold, AA-SBC normally processes registrations. Because the REGISTER dynamic threshold adjusts down to accommodate CPU use, when it is equal to the low watermark, it indicates that CPU usage is very high. In this case, when in-progress registrations are above the dynamic threshold, the system responds as follows: • suppresses—requested, waiting, aged, declined, disconnected, or obsolete REGISTERs. • processes normally—challenging, unauthenticated, or responded REGISTERs. • silently discards—rejected, discarded, or aborted REGISTERs. When in-progress registrations are above the dynamic threshold but the threshold is above the low watermark, the system responds as above except that waiting registrations are processed normally. **Example:** set pending-registrations-low-watermark 2500 Enter a value from 1 to 1,000,000. The default setting is **2000** registrations.

Property name	Description
call-admission-control {enabled | disabled}	Enables or disables call admission control (CAC) on this VSP. The following settings are only applicable if **call-admission-control** is enabled: • cac-max-calls • cac-max-calls-in-setup • cac-min-calls-in-setup • cac-max-number-of-tls • cac-max-tls-in-setup • calls-cpu-limit • call-response-code-at-threshold • call-response-string-at-threshold If disabled, only the more general static-stack-settings **max-number-of-sessions** property controls setup and connection limits. See Admission control for an AOR for specific information on CAC settings applicability for an AOR. **Example:** set call-admission-control enabled The default setting is **disabled**.
cac-max-calls {automatic | *integer*}	Sets the maximum number of concurrent calls allowed on this VSP. You can also set server, gateway, trunk-group, and AOR limits, but the overall simultaneous call limit is set by this property. A call is described as any session established between a UAC and a UAS (proxy) or two UACs and a B2B agent (forwarded). For example, a call between two SIP clients registered via AA-SBC to a Broadsoft server counts as two calls. However, a call between a SIP client registered locally and a media gateway only counts as one. A call consists of the inleg and outleg (end-to-end) connections through AA-SBC. **Example:** set cac-max-calls automatic The default value for this property is **automatic**. See Using automatic values for more information.

Admission Control objects

Property name	Description
calls-high-cpu-threshold *percentage*	Sets an upper threshold, as a percentage, for the SIP process CPU average usage. A dynamic threshold is calculated based on the **max-calls-in-setup** property. When usage exceeds this high threshold (the percentage of max calls), AA-SBC decrements the dynamic threshold until it reaches the value set with the **min-calls-in-setup** property or processing drops below the **calls-high-cpu-threshold** value. If the number of calls in setup exceeds this dynamic threshold, AA-SBC responds to new calls with the configured **call-response-code-at-threshold** and **call-response-string-at-threshold** values. **Example:** set calls-high-cpu-threshold 85 Enter a percentage value from 1 to 100. The default setting is **90** percent.
calls-low-cpu-threshold *percentage*	Sets the low-end threshold, as a percentage, for the CPU average usage, based on the dynamic threshold. When the SIP process CPU falls below the low threshold, AA-SBC increments the dynamic threshold by 20 percent of the **max-calls-in-setup** value until it reaches the **max-calls-in-setup** value. **Example:** set calls-low-cpu-threshold 40 Enter a percentage value from 1 to 100. The default setting is **50** percent.

Property name	Description
cac-max-calls-in-setup {automatic | *integer*}	Sets the maximum number of simultaneous inbound and outbound call legs in setup stage allowed by the CAC. A call leg in setup is much more compute-intensive than established call legs, so this value is more restrictive than the concurrent call leg value. A value of 0 causes AA-SBC to decline all calls and registrations. This value is used as part of the calculation for the initial call dynamic threshold for CPU-based call admission control. Also, if the system dropped any calls in the prior interval, the system increases the dynamic call threshold in 5%. This behavior holds true until calls in setup reaches this maximum value. **Example:** set cac-max-calls-in-setup automatic The default value for this property is **automatic**. See Using automatic values for more information.
cac-min-calls-in-setup *integer*	Sets the minimum number of inbound call legs in setup stage allowed by the CAC. This value is used as part of the calculation for the initial call dynamic threshold for CPU-based call admission control. Also, when the CPU use average exceeds the **calls-cpu-threshold**, the system decrements the dynamic threshold, in 10% increments per span, until it reaches this minimum value. **Example:** set cac-min-calls-in-setup automatic The default value for this property is **automatic**. See Using automatic values for more information.
call-response-code-at-threshold *code*	Sets the numeric code that AA-SBC sends to the caller when it drops a call because the number of calls in setup exceeded the call dynamic threshold. **Example:** set call-response-code-at-threshold 480 The default code is **503** (Service Unavailable).

Property name	Description
call-response-string-at-threshold *string*	Specifies a descriptive text string that AA-SBC sends to the caller when it drops a call because the number of calls in setup exceeded the call dynamic threshold. **Example:** set call-response-string-at-threshold “dynamic threshold exceeded” There is no default setting.
cac-max-number-of-tls {automatic | *integer*}	Sets the maximum number of TLS connections allowed on the VSP. This would include TLS connections for any type of SIP traffic, and includes TLS calls in setup and those that are established. When the system establishes the connection, the **cac-max-tls-in-setup** counter goes down but this counter stays the same. This counter decrements when the connection is broken. **Example:** set cac-max-number-of-tls automatic The default value for this property is **automatic**. See Using automatic values for more information.
cac-max-tls-in-setup {automatic | *integer*}	Sets the maximum number of TLS connections allowed to be in the setup stage at one time. Establishing a TLS connection is very compute-intensive, so this value helps protect AA-SBC from being over-burdened by TLS connections. **Example:** set cac-max-tls-in-setup automatic The default value for this property is **automatic**. See Using automatic values for more information.
sip-stack-pre-auth-timeout *seconds*	Specifies the length of time the system waits for a response from the authentication server. Before processing a SIP message, the system requests user authentication from the server, if required. If the server does not respond in the number of seconds specified by this property, the system discards the message. **Example:** set sip-stack-pre-auth-timeout 45 The default setting is **30** seconds.

Property name	Description
sip-stack-pre-auth-max-pending *integer*	Specifies the number of messages that the system will allow in its queue of pending authentication requests. When the maximum is reached, any new messages arriving will be discarded until the queue level drops below this threshold. **Example:** set sip-stack-pre-auth-max-pendings The default setting is **1024** requests.
route-server-lookup-timeout *seconds*	Specifies the number of seconds the AA-SBC waits for a response from the route server engine (see route-server for more information on route server). When the AA-SBC receives a call, it sends a route-server request (if configured). If this timer expires, the AA-SBC returns a 404 (Not Found) message to the caller (because no route is available). **Example:** set route-server-lookup-timeout 25 The default setting is **30** seconds.
route-server-lookup-max-pending *integer*	Specifies the number of messages that the AA-SBC allows in its queue of pending authorization requests. These could be any type of authorization requests, as configured in the session config authorization object. When the maximum is reached, any new messages arriving will be discarded until the queue level drops below this threshold. Use the authentication-settings **max-outstanding-requests** property to control the number of pending requests at the Diameter server. **Example:** set route-server-lookup-max-pending 2048 The default setting is **1024** requests.

Property name	Description
options-throttling {enabled | disabled}	Specifies whether to do “fast path” processing of OPTIONS messages (of the same session). If this property is **enabled**, when AA-SBC receives an OPTIONS message from the phone, it does a cache lookup. A “hit” to the cache indicates that there was no change to the OPTIONS. In this case, AA-SBC does not need to create a session or go through the policy process again. Instead, it can do just the mandatory SIP processing and respond with a “200 OK.” If the cache does not get hit (indicating that there were changes or that the OPTIONS message was not processed before), AA-SBC caches the new state. The next time the phone contacts AA-SBC, AA-SBC replies and re-establishes the session. When set to **disabled**, AA-SBC processes all OPTIONS messages. **Example:** set options-throttling disabled The default setting is **enabled**.
options-rate-low-watermark *msgsPerSecond*	Specifies the threshold at which the system stops accepting OPTIONS messages and begins replying with a “486 busy” response. This value is measured in messages per second, and is an aggregate threshold for all phones. It is only applicable if the **options-throttling** property is enabled. Use the **options-rate-high-watermark** property to set the threshold at which the system begins silently dropping OPTIONS messages. **Example:** set options-rate-low-watermark 900 The default setting is **800** messages-per-second.

Property name	Description
options-rate-high-watermark *msgsPerSecond*	Specifies the threshold at which the system stops replying with a “486 busy” response to OPTIONS messages and begins silently discarding (dropping) them. This value is measured in messages per second, and is an aggregate threshold for all phones. It is only applicable if the **options-throttling** property is enabled. Use the **options-rate-low-watermark** property to set the threshold at which the system begins responding as busy to OPTIONS messages. **Example:** set options-rate-high-watermark 1100 The default setting is **1000** messages-per-second.
pending-edp-transactions-high-watermark *transactions*	Sets the maximum number of EDP transactions and processes allowed before all new transactions and processes are throttled. (A transaction is the piece of the cycle from sending an OPTIONS message to receiving a response; a process is the sequence of transactions used to determine the expiration time.) **Example:** set pending-edp-transactions-high-watermark 120 Enter a value from 1 to 100,000. The default setting is **100** seconds.
pending-edp-transactions-low-watermark *transactions*	Sets the bottom threshold for throttling in the EDP process. When the number of transactions is between the high and low watermarks, the system throttles new EDP processes, but allows transactions that are part of an existing process to continue. **Example:** set pending-edp-transactions-low-watermark 80 Enter a value from 1 to 100,000. The default setting is **50** seconds.

8. Authentication and authorization objects

Authentication description

AA-SBC uses an authentication cache to help manage REGISTER requests requiring authentication. It is not uncommon to authenticate registrations, but if doing so, may be preferable not to have to re-authenticate with every reregister. You should enable this feature if your authentication server is not capable of handling a high registration rate. If an authentication server cannot keep up with AA-SBC authentication requests, the authentications will start timing out, and the registrations will fail. (AA-SBC can also intelligently suppress registrations using the **registration-throttling** property of the registration-plan route or source-route, or location-service address-of-record objects. This property controls when AA-SBC needs to perform authentication and works at the SIP message level.)

The authentication cache mechanism works in the following manner. When AA-SBC receives an authentication request, it enters the authenticated user name (e.g., user@domain.com) into the cache. The entry remains in the cache for the length of time specified by the **cache-timeout** property. AA-SBC then accepts, without forwarding the request to the authentication provider, all authentication requests for that name that occur within that time period. Note that AA-SBC does not cache the associated password as the underlying authentication system does not allow access to it.

AA-SBC removes entries from the authentication cache from one of the following activities:

- expiration of the **cache-timeout** property timer
- reaching cache capacity (as set by the **cache-max** property)
- the authentication-cache-flush action.

You can use the radius-group **authentication-mode** property to implement load balancing of incoming authentication requests across multiple RADIUS servers.

Authorization settings description

The authorization process results in a return of data that controls where a session should be directed and how it is handled.Using the session configuration authorization object, you select the protocol to use to retrieve that data—either the local session configuration, WSDL, or the LCR engine. When the LCR engine is the chosen target, use the authorization-settings object to limit the number of pending requests. The maximum value set applies only to the LCR (diameter) setting, it is not global.

Authentication and authorization object summary

The following table lists and briefly describes the **authentication-settings** and **authorization-settings** objects. See the following chapter for other objects in the CLI hierarchy:

- Chapter 75, “Virtual system partition (VSP) objects”

Object name	Description
`authentication-settings`	*Secondary object.* Configures queue and cache properties for handling authentication requests.
authorization-settings	*Secondary object.* Configures the limit of outstanding requests for the LCR process through the Diameter server.

authentication-settings

Purpose

This is a secondary object. Configures the properties of the authentication cache. See Authentication description for complete details.

Syntax

```
config vsp authentication-settings
```

Properties

Property name	Description
cache {enabled | disabled}	Specifies whether or not to cache authentication responses. When **enabled**, the system caches the responses, which reduces the number of authentication requests sent to RADIUS servers. If your RADIUS server processing speed is likely to impede registrations, you should enable this property. **Example:** set cache enabled The default setting is **disabled**.
cache-timeout *minutes*	Specifies the length of time, in minutes, that the system keeps an authentication response in its cache. When the system receives an authentication response, it timestamps the packet and keeps it in its authentication cache until the timeout expires. While active, any authentication requests for that user name are responded to by the system. When the timer expires, however, the next request for that user name goes to the authentication server. Be aware that enabling this feature can reduce security. The system does not cache passwords and therefore can not know for certain whether the second request is actually coming from the same user and whether it has the correct password. **Example:** set cache-timeout 90 Enter a value between 1 and 10080; the default setting is **60** minutes.

Property name	Description
cache-max *integer*	Specifies the maximum number of authentication responses (user names) that can be stored in the cache at any one time. When the cache is full, if the RADIUS server sends an Accept in response to a request, the system removes the least-recently-used entry from the cache and replaces it with the new entry. **Example:** set cache-max 1250 Enter a value between 2 and 131072; the default setting is **1000** responses.
max-request-queue-length *integer*	Specifies the maximum number of authentication requests that can be queued to a RADIUS server at any one time. This is the number of requests that are pending and have not yet been submitted. Once this limit is reached, the system fails subsequent requests. If set to 0, the default, the system does not limit the number of requests allowed. Note that the requests handled by this property are those that were not satisfied in the authentication cache. **Example:** set max-request-queue-length 200 Enter a value between 0 and 1000000; the default setting is **0** (disabled).

Property name	Description
auth-reject {immediate | delayed *milliseconds*}	Specifies whether the system should report authentication rejections immediately or after a configurable delay. By default, the system reports the rejects immediately. If set to delayed, the system waits the configured number of milliseconds before sending the rejection back to the endpoint. This setting applies only to authentication rejections returned from the authentication provider, not to failed authentication due to other problems (e.g., timeout). You might want to set a delay, for example, to slow a possible password attack by delaying the response to the sender. Also, by configuring a delay on the system, you can disable the feature on your RADIUS server, preventing delay on successful requests and speeding the authentication process. **Example:** set auth-reject delayed 2000 The default setting is **immediate**. If setting to **delayed**, enter a value between 0 and 60,000.

Property name	Description
combine-duplicates {enabled | disabled}	Sets whether the authentication system detects duplicate requests. If **enabled**, when the authentication system detects duplicate requests it only submits a single request to the RADIUS server, but applies the results (accept, reject, timeout, etc.) to all matching requests. Set this to **enabled** if the following conditions exist in your network: • the RADIUS server is configured to delay Reject messages; • the phones retry the request during the Reject delay, causing another delay; • the RADIUS server is under heavy load. • the **error-mode** property is set to **accept**. Do not enable this feature unless all conditions exist. **Example:** set combine-duplicates enabled The default setting is **disabled**.
error-mode {reject | accept}	Sets the behavior when AA-SBC encounters an authentication error (e.g., timeout, resource allocation failure, queue clip, etc.). By default, AA-SBC rejects these calls. When set to **accept**, any error encountered during authentication results in the authentication attempt succeeding. Set the **combine-duplicates** property to **enabled** if this property is set to **accept** to prevent the system from sending duplicate requests to the RADIUS server. The system sends a log message indicating the event. Use the **show authentication** status provider to display a counter of how many unauthorized calls were accepted. **Example:** set error-mode accept The default setting is **reject**.

authorization-settings

Purpose

This is a secondary object. Configures Diameter request restrictions for the route server process. See route-server for more information on the use of Diameter for route server requests.

Syntax

```
config vsp authorization-settings
```

Properties

Property name	Description
max-outstanding-requests *integer*	Specifies the maximum number of authorization requests that can be queued to a Diameter server at any one time. This is the number of requests that are pending and have not yet been submitted. Once this limit is reached, the system fails subsequent requests. This setting only applies if the authorization object mode property is set to **diameter**. **Example:** set max-outstanding-requests 1024 Enter a value between 32 and 4096; the default setting is **2048**.

9. Autonomous IP object

Autonomous IP description

The autonomous IP objects allows you to define “islands” of IP subnets. Traffic from these subnets can then be excluded from passing through AA-SBC (the system does not anchor the interaction). This might be desirable, for example, for phones within the same office. By implementing this feature, known as “smart anchoring,” you can ensure more efficient processing of media streams within identified subnets.

Note that to implement this feature you must set the **anchoring** property to **auto** in the media object of your session configuration. The auto feature allows AA-SBC to determine whether or not to anchor a call.

Autonomous IP subnets are defined by an IP address and mask using the private-group object.

When a SIP call matches a source and destination subnet in a single autonomous IP location private-group, AA-SBC does not anchor the call unless an explicit policy to do so takes precedence. If both the source and destination locations are unassigned (not in a configured subnet), the media anchoring decision follows the behavior defined in the default session config or other session config.

When hosts are behind the same firewall, AA-SBC, by default, does not anchor the calls. However, you can configure the system so that it does anchor those calls by using the private-group object. Each endpoint behind a firewall has a public and private address. Initially, AA-SBC examines the public address to see whether it is comprised within the address pool defined for a private-group. If there is a match, AA-SBC then examines any private groups configured under the private gateway, and applies the same logic as the non-firewall traffic.

AA-SBC media anchoring decision process

AA-SBC first determines membership in location groups and then executes the decision algorithm to decide whether to perform media anchoring. The caller and callee addresses may be defined within a configured subnet (location) group, in which case the membership is determined. Otherwise, AA-SBC determines location membership through the location database.

A location is identified as a pair consisting of a public and private address. The *public address* is the address of either a private location gateway, if present, or a phone client address. The *private address* is the phone client address, which comes from the Contact header in the REGISTER request. If the location is behind a private location gateway (SIP proxy, firewall, etc.), then the public address is different from the private address.

Determining location group membership

When AA-SBC receives a call from a private gateway, it performs the following processing to determine location group membership:

1. AA-SBC obtains the source location (source public address and private address) by looking up the From URI in the location database.
2. If the public address and private address are the same, AA-SBC checks to see if this address is in a configured location group.
3. If the public address and private address are *not* the same, AA-SBC determines the private location gateway in use and checks to see if this address is in a configured location group.
4. AA-SBC obtains the destination location (destination public address and private address) by looking up the Request URI or the To URI in the location database.
5. If the public address and private address are the same, AA-SBC checks to see if this address is in a configured location group.
6. If the public address and private address are *not* the same, AA-SBC determines the private location gateway in use and check to see if this address is in a configured location group.

Determining whether to anchor

Once AA-SBC establishes the location group, it runs a decision algorithm to determine whether to anchor the call. Source and destination groups are configured with the private-group object. The following summarizes the decision process:

1. If the source group and/or destination group are configured, but are different, then AA-SBC anchors the media;
2. If the source and destination groups are configured and if they are the same, then AA-SBC does not anchor the media;
3. If the source firewall and/or the destination firewall are configured, but are different, then AA-SBC anchors the media;
4. If the source and destination firewalls are configured and are the same, then AA-SBC does not anchor the media
5. If the source and destination public address are the same, then AA-SBC does not anchor the media;
6. In all other cases, AA-SBC anchors the media.

Autonomous IP object summary

The following table lists and briefly describes the **autonomous-ip** objects. See the following chapters for other objects in the CLI hierarchy:

- Chapter 75, “Virtual system partition (VSP) objects”

Object name	Description
`autonomous-ip`	Opens the object from which you configure subnets for smart anchoring.
`private-group`	Enters subnets into a group to establish candidates for smart anchoring.

autonomous-ip

Purpose

Opens the object form which you configure autonomous IP subnets in order to implement smart anchoring. For smart anchoring to take effect, you must set **anchoring** to **enabled** in the media object of your session configuration.

Syntax

```
config vsp autonomous-ip
```

Properties

Property name	Description
treat-tag-as-group {true | false}	Enables tag routing for autonomous IP groups. When set to **true**, AA-SBC treats the final routing tag (the tag with any changes that resulted from a matching session configuration) as an autonomous-ip private-group name. In that way, even if an endpoint does not fall within the subnets defined for the group, if it has a routing-tag that matches the VLAN group name, it is treated as part of the group. **Example:** set treat-tag-as-group true The default setting is **false**.

private-group

Creates a gateway for autonomous private IP groups. Use this object if you are configuring a SIP proxy, a firewall, or something similar that serves as the last hop from a phone client to AA-SBC when the phone client registers. If the last-hop device has multiple public addresses, this object sets the address pool that covers the range of addresses. When AA-SBC receives a call from a public address that is defined within the pool, it does not anchor the call.

Also this object adds subnets to the named group. If both the source and destination addresses are contained in one of the subnet entries in the group, and the **connected** property is set to **true**, no media anchoring applies.

Syntax

```
config vsp autonomous-ip private-group name
```

Property name	Description
subnet *ipAddress*/*mask*	Specifies the subnet(s) that you want to exclude from system anchoring. Within a single group, when both the source and destination are found within one of the configured subnets, the connection is a candidate for smart anchoring. Assuming media anchoring is set to **auto**, the **connected** property (below) is set to **true**, and no policy takes precedence, the system does not anchor the call. You can enter multiple subnet addresses. The system processes the subnets in the order that they appear in the configuration. Use the global move command to re-order the subnets for processing. **Example:** set subnet 192.168.0.0/16 The default setting is **0.0.0.0/32**.

Property name	Description
connected {true | false}	Specifies whether all members of the group can reach each other. If set to **true**, and all members are connected, the system does not anchor the media. **Example:** set connected false The default setting is **true**.
self-connected {true | false}	Specifies whether the system should anchor calls if they appear to have the same public and private IP address. (This may happen, for example, if both phones are behind a device that rewrites SIP headers, such as an ALG, and are NAT’d to the same public IP address.) By default (**true**), the AA-SBC device does not anchor those calls. When two phones are in the same group and have the same IP address, if this property is set to **false**, AA-SBC anchors the call. **Example:** set self-connected false The default setting is **true**.

10. BOOTP client and server objects

BOOTP description

The BOOTP objects allow you to configure the Bootstrap Protocol (BOOTP) client and server settings in a AA-SBC network cluster. *Bootstrap Protocol*, described in RFC 951, is the Internet protocol that allows a network client to learn its own IP address and boot information from a BOOTP server.

In a AA-SBC network cluster, a BOOTP client (drone) requests its own IP address from the BOOTP server (master), as well as the IP address of the BOOTP server itself using the hardware MAC address. The BOOTP server responds to BOOTP client requests over the configured server UDP port.

If a BOOTP session cannot be established between the AA-SBC client and server, BOOTP closes the session across the BOOTP interfaces after 60 seconds.

Note: The BOOTP client and server objects are located in different places in the CLI hierarchy. You configure the client within the box object and the server within the interface object.

BOOTP client object summary

You configure BOOTP clients from the box and cluster configuration objects. The box configuration object allows you to configure the BOOTP client on the locally attached AA-SBC device; the cluster configuration object allows you to configure the BOOTP client on individually numbered (indexed) AA-SBC device in a network cluster.

The following table lists and briefly describes the **bootp-client** objects. See the following chapters for other objects in the CLI hierarchy:

- Chapter 14, “Cluster, box, and interface objects”

Object name	Description
`bootp-client`	Opens the bootp-client configuration object. for editing.

BOOTP server object summary

You configure BOOTP servers on an IP interface.

The following table lists and briefly describes the **bootp-server** objects. See the following chapters for other objects in the CLI hierarchy:

- Chapter 14, “Cluster, box, and interface objects”
- Chapter 77, “VLAN objects”
- Chapter 35, “IP objects”

Object name	Description
`bootp-server`	Opens the bootp-server configuration object for editing.

bootp-client

Purpose

Opens the BOOTP client configuration object on the locally attached AA-SBC, or opens the BOOTP client configuration object on the specified AA-SBC device in a cluster configuration.

Syntax

```
config box bootp-client
config cluster box integer bootp-client
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables BOOTP client services on the system. **Example:** set admin enabled The default administrative state is **enabled**.
client-port eth*X* [*portNumber*]	Sets the system interface over which BOOTP sends requests to the BOOTP server (master). The client (drone) will receive the IP address for this interface from the BOOTP server. **Example:** set client-port eth1 68 The default interface is **eth0**. The default known port number for the BOOTP client port is UDP port **68**. Enter an interface number between 0 and 19, and port number from 1 to 65535.
server-port *portNumber*	Sets the system interface over which BOOTP sends responses to the BOOTP client (drone). The server (master) sends IP addresses over this interface to the BOOTP client interfaces. **Example:** set server-port 67 The default known port number for the BOOTP server port is UDP port **67**. Enter a port number from 1 to 65535.

bootp-server

Purpose

Opens the BOOTP server configuration object on the specified IP interface.

Syntax

```
config cluster box number interface ethX ip name bootp-server
config cluster box number interface ethX vlan number ip name
   bootp-server
config box interface ethX ip name bootp-server
config box interface ethX vlan number ip name bootp-server
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables the BOOTP server on the current system Ethernet or VLAN interface. **Example:** set admin enabled The default setting is **enabled**.
port *portNumber*	Sets the UDP port over which BOOTP sends responses to the BOOTP client (drone). The server (master) sends IP addresses over this port to the BOOTP clients. **Example:** set port 67 The default known server port number for BOOTP is UDP port **67**. Enter a port number in the range of 1 to 65535.

11. Calling Group objects

Calling group description

The calling-group object creates different groups (profiles) that can be referenced through the registration-plan, dial-plan, and session config calling-group-settings objects to control routing of outgoing calls. A calling group allows you to group phones behind a single device such as a PBX or an ATA. The associated lines do not have registration capabilities. This feature allows you to add a dial-plan that points to a calling-group in order to route calls to those devices.

Through calling groups you can also group users. For example, you could create groups of users from an enterprise, from an enterprise department, or from another service provider. It creates a way to segregate routing arbitration, call routing, policy, and normalization based on the user group.

Calling groups are useful when an endpoint is on a dynamic IP address, such as a DSL circuit or shared port adapter. When configured, AA-SBC can learn the address dynamically (from the incoming REGISTER request) and use routes configured specifically for that group instead of being forwarded into the general dial-plan table. Calling groups are not meant for multiple standalone endpoints; instead they are for meant for devices that shelter multiple endpoints behind them.

You must enable **calling-group-routing** in the VSP settings object for dynamic learning and the route and source-route functionality. Calling group routing works as follows.

1. AA-SBC receives a REGISTER request.
2. If the REGISTER matches a configured registration-plan, AA-SBC checks to see if there is an associated calling-group.
3. If there is an associated calling-group, and **calling-group-routing** is enabled, AA-SBC binds the calling group to the IP address of the device that sent the registration.

4. A calling group can have only one associated IP address. If AA-SBC receives a REGISTER for an existing calling group but the IP address is different, it overwrites the known address with the new one.

5. When AA-SBC receives an INVITE, and calling-group-routing is enabled, it checks the IP address of the endpoint against all configured calling groups. If there is a match, AA-SBC performs a dial-plan lookup within the configured calling-group routes and source-routes. If there is not a match, the call is not routed. If calling-group-routing is disabled, AA-SBC uses the routes and source routes within dial-plan to handle the call.

You can also direct incoming calls to a calling group. To do so, set the dial-plan **peer** property to **calling-group** and reference the group you want matching calls routed through.

Calling group object summary

The following table lists and briefly describes the **calling-groups** object. See the following chapters for other objects in the CLI hierarchy:

- Chapter 75, “Virtual system partition (VSP) objects”

Object name	Description
`calling-groups`	Opens the calling-group object, allowing you to create group profiles.
`  group`	Creates a calling group profile.
`  route`	Configures match criteria for calling-group member route selection based on the Request URI.
`    condition-list`	Sets matching conditions for application of the route plan. See Chapter 15, “Condition list objects”.
`  source-route`	Configures match criteria for calling-group member route selection based on the source IP address rather than the Request URI.
`    condition-list`	Sets matching conditions for application of the route plan. See Chapter 15, “Condition list objects”.

calling-groups

Purpose

Opens the calling groups object. From here, you create or edit the group profiles referenced by configured registration-plan entries. Note that you must enable **calling-group-routing** in the VSP settings object to administratively enable the calling-group functionality.

Syntax

```
config vsp calling-groups
```

group

Purpose

Creates a group profile that can then be referenced by the registration plan route and source-route configuration entries. The profile creates a way to segregate routing arbitration, call routing, policy, and normalization based on the user group.

Syntax

```
config vsp calling-groups group name
```

Properties

Property name	Description
admin {enabled | disabled}	Sets the administrative state of the calling group. **Example:** set admin enabled The default setting is **enabled**.
domain *domainName*	Sets the domain to which the calling group belongs. **Example:** set domain voip.companyABC.com There is no default setting.

Property name	Description
group-identity *URI*	Specifies the URI identity associated with this group. This is a descriptive field only. Typically, the identity is the main number for the calling group. It could be exchanged, for example, within a SIP message. **Example:** set group-identity sip:19788235200@companyABC.com There is no default setting.
service-type {provider | internal | external}	Specifies the way in which the system handles INVITE and REGISTER requests and database exchanges. See Service-type definitions for complete descriptions of each option. **Example:** set service-type internal The default setting is **internal**.
`routing-tag` *`string`*	Controls which outbound interface SIP traffic uses. The routing-tag indicates the interface on the system where a SIP message with a matching routing-tag would be forwarded. The SIP message derives its routing-tag from the session config or IP interface classification-tag, depending on the configuration scenario. This property sets the initial routing tag for an AOR in a calling group during registration. If there is a policy match that applies to the group, and that configuration sets a routing tag (with the routing-settings ingress- and egress-classification-tag), the policy setting takes precedence. **Example:** set routing-tag grp1 There is no default setting.
max-number-of-addresses *integer*	Specifies the maximum number of AORs that can be associated to the calling group. **Example:** set max-number-of-addresses 5 Enter a value from 1 to 32; the default setting is **1**.

Property name	Description
unregistered-sender-directive {allow | discard | refuse [*resultCode*] [*resultString*]}	Sets the action the system takes when it receives a packet with an unknown sender in the "From" field of the INVITE packet. Use the **registration-requirement-level** setting in the route or source-route object to define what is considered unknown. Select one of the following actions: • **allow**—the system permits the packet to proceed toward its destination. • **discard**—the system immediately discards the packet. • **refuse**—the system discards the packet but sends a response to indicate having done so. The response includes an error code (default of 400 but you can enter any value between 400 and 699) and an optional description. **Example:** set unregistered-sender-directive refuse 404 "unknown sender" The default setting is **allow**. If you select **refuse**, the default result code is **400**.
inbound-session-config-pool-entry *sessionConfigReference*	Specifies a saved session configuration to apply to calls made from this calling group. **Example:** set inbound-session-config-pool-entry "vsp session-config-pool entry callsFromEast" There is no default setting.
outbound-session-config-pool-entry *sessionConfigReference*	Specifies a saved session configuration to apply to calls made to this calling group. **Example:** set outbound-session-config-pool-entry "vsp session-config-pool entry callsToEast" There is no default setting.

Property name	Description
preference {none | *preference*}	Specifies the preference for the group. The lower the value the higher the preference. If you use the value of **none**, the system uses the preference set in a different part of the configuration, such as the ordered set of arbitration rules in the dial-plan object. **Example:** set preference 10 The default preference setting is **none**.
admission-control {enabled | disabled}	Specifies whether the system applies limitations when forwarding a call from this group. The system tracks the number of concurrent (both incoming and outgoing) active calls for this group. If this property is **enabled**, the system does not forward calls from this group if the limit has been reached and instead sends a “503 Service Unavailable” message. If **disabled**, the system does forward calls from the group. (Set the call limit with the **max-number-of-concurrent-calls** property.) **Example:** set admission-control enabled The default setting is **disabled**.

Property name	Description
emission-control {enabled | disabled}	Specifies whether the system considers upstream server capacity when forwarding a call to the group. The system tracks the number of concurrent (both incoming and outgoing) active calls for the group. If this property is **enabled**, the system does not forward calls to the group if the limit, set with the **max-number-of-concurrent-calls** property, has been reached. Instead, the system sends one of the following messages and drops the call: • If there is one outbound server/UAC/UAS, the system sends a “486 Busy” message, indicating that the route was resolved but that the AOR was unavailable. • If there are multiple outbound server/UAC/UASs and all have reached the maximum concurrent calls threshold, the system sends a “486 Busy” message. • If there are multiple outbound server/UAC/UASs and at least one has not reached the maximum concurrent calls threshold, the return code is determined by the final server that the system attempted to reach. This could be, for example, “486 busy” or a “504 server timeout” if the last server was unresponsive and the transaction timed out. If **disabled**, the system continues to forward calls to the group. **Example:** set emission-control enabled The default setting is **disabled**.
max-bandwidth {unlimited | *kbps*}	Specifies the amount of bandwidth the system allocates to this group. **Example:** set max-bandwidth enabled The default setting is **unlimited**.

Property name	Description
max-number-of-concurrent-calls *integer*	Specifies the maximum number of active incoming and outgoing calls allowed for this group at one time. When this number is reached, the system applies admission and emission control, causing either a decline or busy status until the value drops below the threshold. **Example:** set max-number-of-concurrent-calls 1000 Enter a value between 0 and 1,000,000; the default is **1500** calls. A value of 0 causes the system to decline all calls and registrations.

Property name	**Description**
max-calls-in-setup *integer*	Sets the maximum number of simultaneous inbound and outbound call legs in setup stage that are allowed for the calling group. A call leg in setup is much more compute-intensive than established call legs, so this value is more restrictive than the concurrent call leg value. A value of 0 causes the system to decline all calls and registrations. **Example:** set max-calls-in-setup 50 Enter a value between 0 and 10,000; the default is **30** call legs. A value of 0 causes the system to decline all calls and registrations.
call-rate-limiting {enabled [*callsPerInterval*] [*interval*] [*resultCode*] [*resultString*] | disabled}	*Secondary property.* Limits the number of calls sent to a group within a certain interval. Once this interval is reached, the system hunts for the next available calling group. If there are no available servers, the system returns a response code and message. This feature sets the acceptable arrival rate for incoming calls when is use with **admission-control** and the acceptable set-up rate when used with **emission-control**. If **enabled**, set the number of calls allowed and the measurement interval (in seconds). You can also enter a result code from 400 to 699 and a text string to accompany call rejection if no available calling group is found. **Example:** set call-rate-limiting enabled 50 1 480 “Temporarily unavailable” The default value is **disabled**. If set to **enabled**, the default calls-per-interval is 60, the default interval is 1 second, and the default result is 486, Busy Here.

route

Purpose

Configures AA-SBC to make call routing/forwarding decisions based on information in the Request URI. Use the source-route object to make routing decisions based on the IP packet header or the From URI of the SIP message. With the source-route object a route is selected based on the source while this object selects based on the destination. AA-SBC checks for a source-route match first, then a route match.

The route configuration specifies the portion of the Request URI (dial prefix, domain suffix, condition list criteria) to match on to initiate direction of the call to a particular gateway. If an outgoing call matches the **request-uri-match** value specified in the entry, AA-SBC applies the entry session configuration to the call.

For detailed information on how AA-SBC selects routes, see Finding the most-specific entry and Assigning priority in the dial-plan route description. For details of the session configuration objects, see the Chapter 62, “Session configuration objects” descriptions.

Syntax

```
config vsp calling-groups group name route name
```

Properties

See the dial-plan route object for property descriptions.

source-route

Purpose

Configures AA-SBC to make call routing/forwarding decisions based on information in the IP packet header or the From URI of the SIP message. (Use the route object to make routing decisions based on Request URI information.) With the route object a route is selected based on the destination while this object selects based on the source. AA-SBC checks for a source-route match first, then a route match.

The source-route configuration specifies the portion of the IP header or From URI to match on to initiate direction of the call to a particular gateway (set with the **peer** property). If an outgoing call matches the **source-match** value specified in the entry, AA-SBC applies the entry session configuration to the call.

For detailed information on how AA-SBC selects routes, see Finding the most-specific entry and Assigning priority in the dial-plan route description. For details of the session configuration objects, see Chapter 62, “Session configuration objects”.

Syntax

```
config vsp calling-group group name source-route name
```

Properties

See the dial-plan route object for property descriptions.

12. Carriers objects

Carriers description

The carriers objects configures the elements necessary to execute routing arbitration in an environment with multiple gateways. An enterprise generally hosts only a single PSTN gateway, and routing arbitration can be implemented via the server pool. In a carrier network, every gateway can be a PSTN gateway, and each may offer varying levels of service, cost, and quality. The switch and trunk-group objects are equivalent to an enterprise server pool. They apply routing arbitration to a gateway in order to determine the preferred carrier.

Central to the carrier configuration is the concept of trunks and trunk groups. For a complete description of trunk groups and their relevance, see the IETF working draft, *Representing trunk groups in tel/sip Uniform Resource Identifiers (URIs).*The following paragraphs summarize (and paraphrase) the draft.

In a PSTN, calls are routed over circuits (trunks) between Time Division Multiplexed (TDM) circuit switches. In some cases, multiple trunks (a trunk group) may connect to the same network. The IETF standard uses labels to distinguish the TDM switches in the call routing path. The carrier object, and its subobjects, configure these connections for recognition by AA-SBC.

A *trunk* is “a communication path connecting two switching systems used in the establishment of an end-to-end connection.” A *trunk group* is a “set of trunks...grouped under a common administrative policy for routing...for the establishment of connections within or between switching systems in which all of the paths are interchangeable.”

AA-SBC selects to which carrier it should forward a call based on the configuration of the rate plan and trunk group. The hunt group option allows you to prioritize selections for a carrier, thus implementing a failover strategy.

Normalization in the carriers group

Both the switch and trunk-group objects provide inbound and outbound normalization settings to apply to calls going to or from servers through those types of connections. Use outbound-normalization for calls destined for a server; use inbound-normalization for calls coming received from a server destined for a client. The objects properties are common for servers, gateway, and trunk groups, and are described in Chapter 46, “Outbound and inbound normalization objects”.

Carriers object summary

The following table lists and briefly describes the **carriers** objects. See the following chapter for other objects in the CLI hierarchy:

- Chapter 75, “Virtual system partition (VSP) objects”

Object name	Description
carriers	Opens the object to configure characteristics of the various carrier gateways.
carrier	Creates a carrier configuration.
exchange	Configures the system to recognize the gateways that are part of the carrier network.
switch	Configures the system to recognize the gateways that are part of the carrier network.
network	Sets server socket values. See Chapter 59, “Server objects”, for a full description.
outbound-normalization	Applies normalization settings to outbound calls sent to this gateway. See Chapter 46, “Outbound and inbound normalization objects”, for a full description.
condition-list	Defines the conditions for policy and registration and dial plan matching. See Chapter 15, “Condition list objects”, for a full description.

Object name	Description
inbound-normalization	Applies normalization settings to inbound calls received from this gateway. See Chapter 46, “Outbound and inbound normalization objects”, for a full description.
condition-list	Defines the conditions for policy and registration and dial plan matching. See Chapter 15, “Condition list objects”, for a full description.
rate-plan	Applies a cost basis to calls from this gateway matching the specified Request UR.
trunk-group	Configures the group of trunks associated with a gateway.
outbound-normalization	Applies normalization settings to outbound calls sent to this trunk group. See Chapter 46, “Outbound and inbound normalization objects”, for a full description.
condition-list	Defines the conditions for policy and registration and dial plan matching. See Chapter 15, “Condition list objects”, for a full description.
inbound-normalization	Applies normalization settings to inbound calls received from this trunk group. See Chapter 46, “Outbound and inbound normalization objects”, for a full description.
condition-list	Defines the conditions for policy and registration and dial plan matching. See Chapter 15, “Condition list objects”, for a full description.
rate-plan	Applies a cost basis to calls from this trunk group matching the specified Request URI.
registration-proxy	*Secondary object.* Configures automation of registration database updates between carrier (server) peers that are both proxies.

Object name	Description
`hunt-group`	Sets the criteria the system uses to determine to which carrier it will forward a call.
`class-of-service`	Creates a class of service category.

carriers

Purpose

Opens the carriers object. It is from here that you configure the characteristics of the various carrier gateways.

Syntax

```
config vsp carriers
```

Properties

None

carrier

Purpose

Creates or configures a carrier configuration for implementing multiple PSTN gateways under a single routing arbitration plan.

Syntax

```
config vsp carriers carrier name
```

Properties

Property name	Description
description *string*	Associates a text string with the carrier configuration. The string displays in some event logs and status providers to help identify the target. **Example:** set description carrierWest There is no default setting.
admin {enabled | disabled}	Specifies whether the system uses this carrier in the current session. If **enabled**, the system uses this carrier. If **disabled**, the system does not use this carrier. **Example:** set admin enabled The default setting is **enabled**.
carrier *string*	Associates a text string with a carrier. The string can later be used to group and categorize servers. **Example:** set carrier server1 There is no default setting.

exchange

Purpose

Configures the system to recognize the gateways that are part of the carrier network.

Syntax

```
config vsp carriers carrier name exchange name
```

Properties

Property name	Description
description *string*	Associates a text string with the carrier configuration. The string displays in some event logs and status providers to help identify the target. **Example:** set description carrierWest There is no default setting.
admin {enabled | disabled}	Specifies whether the system uses this carrier in the current session. If **enabled**, the system uses this carrier. If **disabled**, the system does not use this carrier. **Example:** set admin enabled The default setting is **enabled**.
domain *`domainName`*	Identifies a domain to be used by the system for server normalization. In cases where the server is associated with: • a single domain—enter that domain. • multiple domains—enter one of the domain names. • no domain—enter another valid domain on the system. (This might be the case with a PSTN gateway for example.) Also, you must set this property if you enable the settings **local-directory-based-user-services** property without configuring the **directory** property (to assign a directory to a server). Set this domain name to match user SIP addresses to the appropriate server (by use of the domain). **Example:** set domain voip.companyABC.com There is no default setting.

Property name	Description
routing-tag *string*	Controls which outbound interface SIP traffic uses. The routing-tag indicates the interface on the server where a SIP message with a matching routing-tag would be forwarded. The SIP message derives its routing-tag from the session config or IP interface classification-tag, depending on the configuration scenario. This property sets the initial routing tag for a server. If there is a policy match that applies to the server, and that configuration sets a routing tag (with the routing-settings ingress- and egress-classification-tag), the policy setting takes precedence. **Example:** set routing-tag lcs1 There is no default setting.
routing-setting *attributes*	Sets attributes of the carrier/server. See Routing-setting definitions for a description of each option. **Example:** set routing-setting auto-tag-match+auto-domain-match The default setting is **normalization** and **outbound-association**.
domain-alias *`domainName`*	Sets the system to recognize an alias domain as the domain in which the carrier resides. You can enter as many aliases as you choose. **Example:** set domain-alias eng.companyABC.com There is no default setting.
domain-subnet *`ipAddress/mask`*	Sets the IP subnets serviced by this carrier. You can enter as many subnets as you choose. **Example:** set domain-subnet 1.2.3.4/16 There is no default setting.
service-type {provider | internal | external}	Specifies the way in which the system handles INVITE and REGISTER requests and database exchanges. See Service-type definitions for complete descriptions of each option. **Example:** set service-type internal The default setting is **provider**.

Property name	Description
failover-detection {none | auto | ping | register}	Determines the method to use to detect a when a upstream gateway is unavailable (and when it has resumed availability). Select either: • **none**—the system does no checking; the gateway always appears available, even when down. • **auto**—the system uses an internal algorithm to count transaction failures. Use this setting in the case where a gateway does not respond to SIP OPTION messages. When the gateway reaches the failure threshold (set with the **dead-threshold** property), the system changes the state to DOWN and sends no further requests. The fallback timer (set with the **dead-fallback-interval** property) activates. When the timer expires, the system decrements the gateway dead count by one and can again send requests to the server. If it receives no response, the system again increments the count and reaches the threshold, restarting the process. If the gateway responds, the system decrements the dead count again, until the count reaches 0. Note that you must also enable the vsp **auto-server-fail-detection** property (which acts like a master switch) when using this option. continued

Property name	Description
`failover-detection` *continued*	• **ping**—the system uses the SIP ping utility to check gateway availability. When enabled, SIP ping sends SIP OPTION messages to each gateway at an interval defined in the ping-interval property. If the gateway is not operational, the system switches to local mode if pstn-backup is checked (**routing-setting** property), or to unavailable mode if pstn-backup is not checked. When the system is again able to successfully ping the gateway, it reverts to provider mode. When the gateway reaches the failure threshold (set with the **dead-threshold** property), the system changes the state to DOWN and sends no further requests. • **register**—the system determines gateway availability by sending a REGISTER request to the gateway. If there is no response from the provider, the server is assumed down. Note that you must configure a user for the server, with the appropriate password. (See the **user** and **password-tag** properties). **Example: set failover-detection auto** The default setting is **none**.
ping-interval *seconds*	Sets the number of seconds between ping packets sent between the system and the carrier gateways and the timeout before a DNS lookup failure retry. **Example:** set ping-interval 30 The default setting is **10** seconds.
dead-threshold *retransmissions*	Specifies the number of transaction failures (and resulting retransmissions) a gateway can experience before the state is changed to DOWN. This threshold is used in the **auto** or **ping** option of the **failover-detection** property. It is also used to determine the number of DNS lookup retries the system attempts before giving up. **Example:** set dead-threshold 8 Enter a value from 1 to 255; the default setting is **4**.

Property name	Description
dead-fallback-interval *seconds*	Sets the fallback timer for the gateway. During this period, the system does not send REGISTER or INVITES to the down gateway. After the timer expires, the system decrements the **dead-threshold** by 1. This timer is used in the auto option of the **failover-detection** property. It is also used to determine when the system should re-attempt to resolve a failed DNS lookup of a peer. **Example:** set dead-fallback-interval 450 The default setting is **300** seconds.
gateway-flush {enabled | disabled}	Specifies whether to flush any known carrier gateways from its gateway status table and relearn their gateway status (specifically host address) from static configuration and DNS. If **enabled**, the system uses the timer set with the **gateway-age** parameter to determine how often to flush the data. If **disabled**, the system does not remove carrier gateway data from the configuration. **Example:** set gateway-flush enabled The default setting is **disabled**.
handle-3xx-locally-routing-lookup {enabled | disabled}	Specifies whether the system should do a dial plan lookup on the REQUEST URI of a newly generated INVITE based on a 302 response received from this carrier. This property works in conjunction with the **handle-3xx-locally** property of the sip-settings session config object. If that property is enabled, the system generates a new INVITE when it receives a 3xx response. The system puts the contents of the CONTACT field in the REQUEST URI of the new INVITE. You should **disable** this feature if your server is configured to explicitly forward the message to a specified third sever. Set this to **enabled** if the message is coming from an endpoint with instructions to forward the message to a different AOR. **Example:** set handle-3xx-locally-routing-lookup disabled The default setting is **enabled**.

Property name	Description
unregistered-sender-directive {allow | discard | refuse [*resultCode*] [*resultString*]}	Sets the action the system takes when this carrier receives a packet with an unknown sender in the “From” field of the INVITE packet. Use the **registration-requirement-level** setting in the route or source-route object to define what is considered unknown. Select one of the following actions: • **allow**—the system permits the packet to proceed toward its destination. • **discard**—the system immediately discards the packet. • **refuse**—the system discards the packet but sends a response to indicate having done so. The response includes an error code (default of 400 but you can enter any value between 400 and 699) and an optional description. **Example:** set unregistered-sender-directive refuse 404 “unknown sender” The default setting is **allow**. If you select **refuse**, the default result code is 400.
sip-identity *sipURI*	*Secondary property.* Specifies the URI for this carrier. Use this property if you swap registration databases with peers. When a download occurs, each entry is marked with the carrier-identity from which the entry was learned, if known. In that way, when the system next downloads the database, those entries learned from the same identity as this carrier-identity are not downloaded. **Example: set sip-identity sip:provider.com** There is no default setting.

Property name	Description
call-hunting-type {none | sequential | parallel}	*Secondary property.* Determines the order or method in which the system forwards the call to the next-hop gateway. • **none**—the system forwards the call to the latest binding for the Request URI. • **sequential**— if there are two or more gateways configured for a carrier, the system first tries the primary and then the secondary. • **parallel**—when the system receives a call, it creates two call legs and forwards to both the primary and secondary gateway. When one gateway responds, the system disconnects the call with the other gateway. **Example:** set call-hunting-type none The default setting is **sequential**.
loop-detection {strict | tight | loose}	*Secondary property.* Sets the level at which the system enforces call routing loop detection to each gateway. Select one of the following: • **strict**—if the system receives a call from a SIP proxy, and a DNS or dial-plan lookup resolves that the source and destination address are the same, the system drops the call. • **tight**—if the system finds the source and destination address, transport protocol, and port to be the same, it drops the call. • **loose**—the system uses standard SIP loop detection (based on the VIA header). When the system finds its own address in the list of SIP proxies traversed, it allows the packet through. **Example:** set loop-detection strict The default setting is **tight**.

Property name	Description
gateway-age *seconds*	*Secondary property.* Sets the frequency with which the system flushes carrier gateway data from its gateway status table and relearns it from DNS (or the static configuration). This value is only applicable if the **gateway-flush** property is **enabled**. **Example:** set gateway-age 129600 The default setting is **86,400** seconds.
peer-max-interval *seconds*	*Secondary property.* Specifies the value the system writes to the max-interval setting for a carrier gateway. When doing registration delegation, the system changes the expiration value in the REGISTER request to the specified **max-interval** when delegating it to the upstream gateway. The system saves the expiration value recorded in the 200 OK from the upstream gateway to its location cache. If you enter 0, the peer value remains. **Example:** set peer-max-interval 0 The default setting is **86,400** seconds.
peer-min-interval *seconds*	*Secondary property.* Specifies the value the system writes to the min-interval setting for a carrier gateway. When doing registration delegation, the system changes the expiration value in the REGISTER request to the specified **min-interval** when delegating it to the upstream gateway. The system saves the expiration value recorded in the 200 OK from the upstream gateway to its location cache. If you enter 0, the peer value remains. **Example:** set peer-min-interval 0 The default setting is **3600** seconds.
registration-request-timeout *seconds*	*Secondary property.* Specifies the number of seconds the system waits for a response after sending a REGISTER request to this carrier. If the system does not receive a response within the configured time, it sends, to the endpoint, notification that service is not available. **Example:** set registration-request-timeout 8 The default setting is **10** seconds.

Carriers objects

Property name	Description
user `string`	*Secondary property.* Assigns a user name that the system must supply when challenged by the carrier gateway (typically, the phone number the carrier expects). Enter the string expected by the carrier, do not create it here. The user name and password-tag (below) are used for authentication between the system and carrier. **Example:** set user 9785551212 There is no default setting. This string must match the username configured on the server.
password-tag `string`	*Secondary property.* Specifies the tag associated with the shared secret used to authenticate transactions between the system and carrier. This is the tag associated with the password that the system must supply when challenged by the carrier. See Understanding passwords and tags for information on the AA-SBC two-part password mechanism. **Example:** set password-tag secure There is no default setting. This password associated with this tag must match the password configured on the carrier.
dialog-failover	When enabled, the **dialog-failover** setting forces AA-SBC to check the state of the destination SIP server before sending messages. If the destination server is down, the calls are routed to the next configured (and available) backup server. **Note:** For dialog-failover to work, the **failure-detection** property must be set to *auto*, *ping*, or *register* in the **servers** and/or **exchange** objects. When **dialog-failover** is set to disabled, any calls in progress at the time of the failure will be retried at the original destination server until the configured timeout settings have expired. **Example**: **set dialog-failover enabled** The default setting is **disabled**.

switch

Purpose

Configures AA-SBC to recognize the gateway switches that are part of the carrier network. Configure each switch individually. These properties are similar to the configuration attributes that apply to the gateway trunk-group configuration. The gateway attributes only need be configured if the gateway does not have trunks associated with it.

The switch object allows you to configure normalization plans for outgoing and incoming calls using this gateway switch. See Chapter 46, “Outbound and inbound normalization objects”, for a full description of the gateway object outbound-normalization and inbound-normalization subobjects.

Syntax

```
config vsp carriers carrier name exchange name switch name
```

Properties

Property name	Description
description *string*	Associates a text string with the switch configuration. The string displays in some event logs and status providers to help identify the target. **Example:** set description carrierWest There is no default setting.
endpoint *string*	Associates a text string with the switch. The string can later be used to group and categorize gateway servers. **Example:** set endpoint server1 There is no default setting.
host *server*	Specifies the name or IP address of the gateway switch. Enter a host name or IP address. **Example:** set host 192.168.10.10 There is no default setting.

Property name	**Description**
transport {any | UDP | TCP | TLS}	Specifies the protocol used by the switch. **Example:** set transport any The default protocol is **UDP**.
port *portNumber*	Specifies the port used by the switch for SIP traffic. **Example:** set port 3333 The default setting is port **5060**.
local-port *portNumber*	Sets a port number for the system to use in the Contact header, and Via header, and source port when it sends a Register request (and subsequent SIP messages) to an upstream server. The server caches the binding and includes the local-port when contacting the system. Additionally, the server can be configured to send SIP messages to this particular local-port without prior registration from the system. With local-port configured, the system can tell: • to which server in the server pool to forward a call • which server in the server pool it received the call from, when the server sends SIP message to this local port. Using this property allows you to group traffic based on the local port number. For example, if there are multiple domains from a single physical server, the port will indicate which domain should receive the call. Or, if there is a distinct pair of physical servers to protect traffic for a domain, the Eclipse can fail over to the right backup server (in case of primary failure) for this particular domain. **Example:** set local-port 5050 There is no default setting.

Property name	Description
connection-role {initiator | responder}	Specifies the way the switch behaves in establishing a TCP/TLS connection. If set to: • **initiator**—the gateway server opens up a connection to its peer(s) on startup. It monitors the connection and retries if the connection fails. (Use the **connection-retry-interval** property to set the frequency.) • **responder**—the gateway only opens a TCP/TLS connection if it receives SIP traffic or if **failover-detection** is set to **ping** in the server object. **Example:** set connection-role responder The default setting is **initiator**.
connection-retry-interval *seconds*	Specifies the number of seconds the system waits between attempts to open a TCP or TLS connection. This value is only meaningful if the **connection-role** property is set to **initiator**. If set to **responder**, the value is ignored. **Example:** set connection-retry-interval 10 The default setting is **5** seconds.
handle-response *code* {try-next-peer | try-next-route | forward}	Specifies the action the system should take when it receives a specific response code from this switch. Enter a code, and set a handling pattern: • **try-next-peer**—the system forwards the message to the next gateway within this carrier network. • **try-next-route**—the system forwards the message to the route that is the next most-specific. • **forward**—the system returns the response to the originator of the message. **Example:** set handle-response 404 try-next-route There is no default value. Enter a response code between 400 and 999. The default setting for the handling pattern is **try-next-peer**.

Property name	Description
admission-control {enabled | disabled}	Specifies whether the system considers downstream gateway capacity when forwarding a call. The system tracks the number of concurrent calls for each gateway switch. If this property is **enabled**, the system does not forward calls if the switch limit has been reached and instead sends a “503 Service Unavailable” message. If **disabled**, the system does forward calls. (Set the call limit with the **max-number-of-concurrent-calls** property.) **Example:** set admission-control enabled The default setting is **disabled**.
emission-control {enabled | disabled}	Specifies whether AA-SBC considers upstream server capacity when forwarding a call to the gateway switch. the system tracks the number of concurrent (both incoming and outgoing) active calls for the switch. If this property is **enabled**, the system does not forward calls to the gateway if the limit, set with the **max-number-of-concurrent-calls** property, has been reached. Instead, the system sends one of the following messages and drops the call: • If there is one outbound server/UAC/UAS, the system sends a “486 Busy” message, indicating that the route was resolved but that the AOR was unavailable. • If there are multiple outbound server/UAC/UASs and all have reached the maximum concurrent calls threshold, the system sends a “486 Busy” message. • If there are multiple outbound server/UAC/UASs and at least one has not reached the maximum concurrent calls threshold, the return code is determined by the final server that the system attempted to reach. This could be, for example, “486 busy” or a “504 server timeout” if the last server was unresponsive and the transaction timed out. If **disabled**, the system continues to forward calls to the switch. **Example:** set emission-control enabled The default setting is **disabled**.

Property name	Description
max-number-of-concurrent-calls *integer*	Specifies the number of calls allowed on the switch at one time. When this value is reached, the switch will not accept calls until the value drops. This option is only applicable if you have set the **emission-control** property to enabled. **Example:** set max-number-of-concurrent-calls 1500 Enter a value between 0 and 1,000,000; the default is **1000** calls.
max-calls-in-setup *integer*	Sets the maximum number of simultaneous inbound and outbound call legs in setup stage that are allowed by the switch. A call leg in setup is much more compute-intensive than established call legs, so this value is more restrictive than the concurrent call leg value. A value of 0 causes the system to decline all calls and registrations. **Example:** set max-calls-in-setup 50 Enter a value between 0 and 10,000; the default is **30**.
trunk-token *string*	Sets the RFC-required trunk group identifier label. Change this value only if your PSTN gateway uses a different trunk token. **Example:** set trunk-token tgrp The default setting is **tgrp**.
preference {none | *preference*}	*Secondary property.* Specifies the preference for the switch. The lower the value the higher the preference. If you use the value of **none**, the system uses the preference set in a different part of the configuration, such as the option settings in the hunt-group object or preference in trunk-group. If the preference is set in multiple places, the system considers them in the following order: 1. option settings in the hunt-group 2. gateway preference (this setting) 13. preference in trunk-group **Example:** set preference 10 The default preference setting is **none**.

Property name	Description
call-routing-on {request-uri | to-uri}	*Secondary property.* Sets the portion of the SIP header that the system uses in its dial-plan lookup. By default, the dial-plan lookup applies to the Request URI. **Example:** set call-routing-on to-uri The default setting is **request-uri**.
max-bandwidth {unlimited | *kbps*}	*Secondary property.* Specifies the amount of bandwidth the system allocates to a switch. For a SIP server, the default value is **unlimited** or the server uplink bandwidth. For example, if the uplink is GigE, then bandwidth is 1 million kbps. Set a specific bandwidth if you are using, for example, a TDM trunk or PSTN gateway with limited bandwidth. For a PSTN trunk, the usual capacity is DS0 (64 kbps bandwidth). If a gateway has 8 trunks, then the gateway has 512 kbps bandwidth. **Example:** set max-bandwidth 512 The default setting is **unlimited**.
call-rate-limiting {enabled [*callsPerInterval*] [*interval*] [*resultCode*] [*resultString*] | disabled}	*Secondary property.* Limits the number of calls sent to the switch within a certain interval. Once this interval is reached, AA-SBC hunts for the next available switch. If there are no available servers, the system returns a response code and message. This feature sets the acceptable arrival rate for incoming calls when is use with **admission-control** and the acceptable set-up rate when used with **emission-control**. If **enabled**, set the number of calls allowed and the measurement interval (in seconds). You can also enter a result code from 400 to 699 and a text string to accompany call rejection if no available switch is found. **Example:** set call-rate-limiting enabled 50 1 480 “Temporarily unavailable” The default value is **disabled**. If set to **enabled**, the default calls-per-interval is 60, the default interval is 1 second, and the default result is 486, Busy Here.

Property name	Description
max-number-of-registrations *value*	*Secondary property.* Specifies the maximum number of registrations that can be active on this switch at any one time. This property is used in conjunction with the **server-registration-balance** property of the VSP settings object to implement registration load balancing. **Example:** set max-number-of-registrations 1500 The default setting is **1000** registrations.
max-registrations-in-progress *value*	*Secondary property.* Specifies the number of registrations or authentication requests per second that the system forwards to the switch. Use this property as a flow control mechanism to control the system, which can process registrations much more quickly than the gateway server. To set this, you must know the capability of your server. You also must enable the **server-registration-balance** property of the VSP settings object. When a register is delegated/forwarded/tunneled to the switch, the system increments a cluster-wide server counter. When the counter reaches this threshold, the system handles subsequent registrations. It responds with “200 OK,” but sets a brief expiration, causing the phone to reregister almost immediately. **Example:** set max-registrations-in-progress 600 Enter a value between 0 and 10,000; the default setting is **300** registrations. A value of 0 causes the system to decline all calls and registrations.

Property name	Description
flat-rate {free | *cents* | not-available}	*Secondary property.* Sets a default rate if the call does not match any of the criteria set forth in the rate plan. This setting does not effect billing in any way, but instead determines the switch selection the system makes. Specifies the cost per call on this gateway switch: • **free**—there is no cost associated. • **cents**— the cost is equal to the value entered. • **not-available**—the system does not select this gateway (cost is too high). **Example:** set flat-rate free The default setting is **not-available**.
external-outbound-normalization {no | yes *server*}	*Secondary property.* Specifies whether the system should perform external normalization on outbound call legs. Enter the host name of your calling plan server. **Example:** set external-outbound-normalization yes ITALKBB-EGR5 The default setting is **no**.
external-inbound-normalization {no | yes *server*}	*Secondary property.* Specifies whether the system should perform external normalization on inbound call legs. Enter the host name of your calling plan server. **Example:** set external-inbound-normalization yes PT1-INGRESS The default setting is **no**.

rate-plan

Purpose

Configures a switch-specific (or trunk-group specific) rate structure. You can select the portions of the call header to match on and then apply cost basis plans accordingly.

A rate plan can be applied to a gateway switch or to a trunk group within a gateway. It sets the rate for calls, which is then used by AA-SBC in the gateway or trunk selection process. (It does not effect actual billing.)

Syntax

```
config vsp carriers carrier name exchange name switch name rate-plan
    name
config vsp carriers carrier name exchange name switch name trunk-group
    name rate-plan name
```

Properties

Property name	Description
request-uri-match *type string*	Specifies what to match in the USER and/or HOST fields of the SIP header in order for the system to apply the rate plan to calls containing the prefix. Select the type of match to make, and then enter a string to match on. **Example:** set request-uri-match phone-prefix 19788235 7 There is no default setting. The default minimum digits is 1.

Property name	Description
rate *dialTimeZoneReference* {free | *cents* | not-available}	References a dial-time-zone configuration and sets a rate for calls matching that defined time zone. The rate is used by the system in its gateway selection process. Select either: • **free**—there is no cost associated. • **cents**— the cost per minute for a call, equal to the value entered. • **not-available**—the system does not select this gateway (typically because the cost is too high). **Example:** set rate “vsp carriers dial-time-zone east” 25 There is no default setting for the dial-time-zone reference.
flat-rate {free | *cents* | not-available}	*Secondary property.* Sets a default cost per call if the call matches the request-uri-match setting but does not match a dial-time-zone set forth in the **rate** property. Specify the cost per call on this gateway: • **free**—there is no cost associated. • **cents**— the cost per minute for a call, equal to the value entered. • **not-available**—the system does not select this gateway (typically because the cost is too high). **Example:** set flat-rate free The default setting is **not-available**.

trunk-group

Purpose

Specifies the configuration for the group of trunks (circuits) associated with a gateway. (These properties are similar to the configuration attributes that apply to the carrier switch configuration. The gateway attributes only need be configured if the gateway does not have trunks associated with it.)

AA-SBC uses a trunk group, a group of trunks that connects to the same target switch or network, to route calls to a PSTN gateway over specific circuits between TDM circuit switches. It is through the dial-plan arbiter configuration that AA-SBC determines over which trunk group of a particular carrier to route a call.

When AA-SBC routes a call to a specific trunk, it appends a tag to indicate to the carrier over which the call should be transmitted. For example, if AA-SBC routes a call to trunk “china,” it appends **tgrp=china** to the Request URI so that the provider gateway correctly transmits the call to trunk-group china. (The peer trunk setting, in the dial-plan route object, specifies over which trunk to route the call.)

The trunk-group object allows you to configure normalization plans for outgoing and incoming calls using this trunk. See Chapter 46, “Outbound and inbound normalization objects”, for a full description of the this trunk-group object outbound-normalization and inbound-normalization subobjects.

Syntax

```
config vsp carriers carrier name exchange name switch name trunk-group
   name
```

Properties

Property name	Description
description *string*	Associates a text string with the trunk-group configuration. The string displays in some event logs and status providers to help identify the target. **Example:** set description carrierWest There is no default setting.
trunk-tag *string*	Assigns a string to the trunk-group. The system appends the tag to the Request URI so that the provider can transmit the call to the correct trunk group. Note that this tag must be consistent with the tag configured at the provider end. **Example:** set trunk-tag china There is no default setting.
handle-response *code* {try-next-peer | try-next-route | forward}	Specifies the action the system should take when it receives a specific response code from this trunk group. Enter a code, and set a handling pattern: • **try-next-peer**—the system forwards the message to the next circuit associated with this gateway. • **try-next-route**—the system forwards the message to the route that is the next most-specific. • **forward**—the system returns the response to the originator of the message. **Example:** set handle-response 404 try-next-route There is no default value. Enter a response code between 400 and 999. The default setting for the handling pattern is **try-next-peer**.

Property name	Description
admission-control {enabled | disabled}	Specifies whether the system considers downstream trunk group capacity when forwarding a call. The system tracks the number of concurrent calls for each trunk group. If this property is **enabled**, the system does not forward calls if the trunk group limit has been reached and instead sends a “503 Service Unavailable” message. If **disabled**, the system does forward calls. (Set the call limit with the **max-number-of-concurrent-calls** property.) **Example:** set admission-control enabled The default setting is **disabled**.
emission-control {enabled | disabled}	Specifies whether AA-SBC considers upstream server capacity when forwarding a call to the trunk group. AA-SBC tracks the number of concurrent (both incoming and outgoing) active calls for the group. If this property is **enabled**, the system does not forward calls to the trunk group if the limit, set with the **max-number-of-concurrent-calls** property, has been reached. Instead, the system sends one of the following messages and drops the call: • If there is one outbound server/UAC/UAS, the system sends a “486 Busy” message, indicating that the route was resolved but that the AOR was unavailable. • If there are multiple outbound server/UAC/UASs and all have reached the maximum concurrent calls threshold, the system sends a “486 Busy” message. • If there are multiple outbound server/UAC/UASs and at least one has not reached the maximum concurrent calls threshold, the return code is determined by the final server that the system attempted to reach. This could be, for example, “486 busy” or a “504 server timeout” if the last server was unresponsive and the transaction timed out. If **disabled**, AA-SBC continues to forward calls to the trunk group. **Example:** set emission-control enabled The default setting is **disabled**.

Property name	Description
max-bandwidth {unlimited | *kbps*}	*Secondary property.* Specifies the amount of bandwidth the system allocates to a trunk group. The default value is **unlimited** or the trunk uplink bandwidth. For example, if the uplink is T1, then bandwidth is 1.544 Mbps. Set a specific bandwidth if you are using, for example, a TDM trunk or PSTN gateway with limited bandwidth. For a PSTN trunk, the usual capacity is DS0 (64 kbps bandwidth). If a gateway has 8 trunks, then the gateway has 512 kbps bandwidth. A trunk-group can be configured max bandwidth AA-SBC can utilize. If an arbiter seeks to balance the load (seeks the least loaded trunk-group) for a call, then the available bandwidth is checked against all candidate trunk-groups. The trunk-group with the most available bandwidth wins the election. **Example:** set max-bandwidth 512 The default setting is **unlimited**.
max-number-of-concurrent-calls *integer*	Specifies the number of calls allowed on the trunk group at one time. When this value is reached, the trunk group will not accept calls until the value drops. This option is only applicable if you have set the **emission-control** property to enabled. **Example:** set max-number-of-concurrent-calls 1500 Enter a value between 0 and 1,000,000; the default is **1000** calls.
max-calls-in-setup *integer*	Sets the maximum number of simultaneous inbound and outbound call legs in setup stage that are allowed for the trunk group. A call leg in setup is much more compute-intensive than established call legs, so this value is more restrictive than the concurrent call leg value. A value of 0 causes AA-SBC to decline all calls and registrations. **Example:** set max-calls-in-setup 50 Enter a value between 0 and 10,000; the default is **30**.

Property name	Description
preference {none | *preference*}	*Secondary property.* Specifies the preference for the trunk group. The lower the value the higher the preference. If you use the value of **none**, the system uses the preference set in a different part of the configuration, such as the option settings in the hunt-group object or preference in switch. If the preference is set in multiple places, the system considers them in the following order: 1. hunt-group `option setting` 2. switch `preference setting` `14.this preference` If an arbiter seeks the most preferred trunk-group for a call, then the preference value is checked against all candidate trunk-groups. The trunk-group with the lowest preference wins the election. **Example:** set preference 10 The default preference setting is **none**.
max-bandwidth {unlimited | *kbps*}	*Secondary property.* Specifies the amount of bandwidth the system allocates to this trunk group. For a SIP server, the default value is **unlimited** or the server uplink bandwidth. For example, if the uplink is GigE, then bandwidth is 1 million kbps. Set a specific bandwidth if you are using, for example, a TDM trunk or PSTN gateway with limited bandwidth. For a PSTN trunk, the usual capacity is DS0 (64 kbps bandwidth). If a gateway has 8 trunks, then the gateway has 512 kbps bandwidth. **Example:** set max-bandwidth 512 The default setting is **unlimited**.

Property name	**Description**
call-rate-limiting {enabled [*callsPerInterval*] [*interval*] [*resultCode*] [*resultString*] | disabled}	*Secondary property.* Limits the number of calls sent to a trunk group within a certain interval. Once this interval is reached, AA-SBC hunts for the next available trunk group. If there are no available servers, the system returns a response code and message. This feature sets the acceptable arrival rate for incoming calls when is use with **admission-control** and the acceptable set-up rate when used with **emission-control**. If **enabled**, set the number of calls allowed and the measurement interval (in seconds). You can also enter a result code from 400 to 699 and a text string to accompany call rejection if no available trunk group is found. **Example:** set call-rate-limiting enabled 50 1 480 “Temporarily unavailable” The default value is **disabled**. If set to **enabled**, the default calls-per-interval is 60, the default interval is 1 second, and the default result is 486, Busy Here.
flat-rate {free | *cents* | not-available}	*Secondary property.* Sets a default rate if the call does not match any of the criteria set forth in the rate plan. This setting does not effect billing in any way, but instead determines the gateway selection the system makes. Specifies the cost per call on this trunk group: • **free**—there is no cost associated. • **cents**— the cost is equal to the value entered. • **not-available**—the system does not select this trunk group (cost is too high). **Example:** set flat-rate free The default setting is **not-available**.

Property name	Description
external-outbound-normalization {no | yes *server*}	*Secondary property.* Specifies whether the system should perform external normalization on outbound call legs. Enter the host name of your calling plan server. **Example:** set external-outbound-normalization yes ITALKBB-EGR5 The default setting is **no**.
external-inbound-normalization {no | yes *server*}	*Secondary property.* Specifies whether the system should perform external normalization on inbound call legs. Enter the host name of your calling plan server. **Example:** set external-inbound-normalization yes PT1-INGRESS The default setting is **no**.

registration-proxy

Purpose

Secondary object. Configures automation of registration database updates between carrier (server) peers that are both proxies. When AA-SBC acts as a proxy, it is able to supply the credentials needed for authentication challenges. It maintains a location service database to store SIP caller location (address-of-record) information. This database can be updated via AA-SBC registration service, static address-of-records (AORs), and/or configured policies. To ensure that peer AA-SBC devices have and use the same database, set the properties of this object.

Syntax

```
config vsp carriers carrier name registration-proxy
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables this proxy registration configuration. If enabled, the server applies these characteristics to sessions with its configured peers. If disabled, these characteristics are inactive. **Example:** set admin disabled The default setting is **enabled**.
request-download {no | yes *interval*}	Automates the download of the registration database from a peer. (Peers are identified in the server configuration.) If set to yes, the peer AA-SBC device downloads the database to this server with the frequency set in the request interval of this property. In addition, it copies the interval to the expiration time in the REGISTER requests forwarded to peers. If set to no, downloads do not occur automatically. AA-SBC only learns of new or changed AORs through REGISTER requests. **Example:** set request-download yes 1080 The default setting is **no**. If set to yes, the default interval is 1440 minutes (24 hours).

hunt-group

Purpose

Sets the criteria AA-SBC uses to determine to which carrier it will forward a call.

A hunt group is a bank of configured carrier trunks. AA-SBC applies the relevant dial-plan arbiter rules to each entry in the hunt-group and, based on those rules, calculates the most preferred carrier for a call. In routing a call, if one trunk-group fails, then AA-SBC hunts for the next available trunk-group in the hunt-group.

The dial-plan route object peer property configures a dial-plan to point to a variety of peers. For example, a carrier gateway with a number of associated trunk groups can be considered as a hunt-group, as can a carrier. An enterprise server with a server pool can also be considered as a hunt-group.

When AA-SBC receives a call, it resolves the subscriber's arbiter. If the resulting arbiter uses the best match, then AA-SBC looks up in the call routing table for the best match. That best match points to a hunt group. The arbiter then applies all its rules to all trunk-groups in that hunt group.

If the resulting arbiter is configured to select all matches, AA-SBC performs a call routing table lookup. For each match, and each associated hunt group, the arbiter applies all rules to the trunk-group(s) in that hunt group. AA-SBC then compares the calculation results between hunt-groups and selects the best.

Syntax

```
config vsp hunt-group name
```

Properties

Property name	Description
admin {enabled | disabled}	Sets the administrative state of the hunt group configuration. **Example:** set admin disabled The default setting is **enabled**.
option {trunk “*trunkReference”* | switch “*gatewayReference”* | exchange “*exchangeReference”* | carrier “*carrierReference”* | server “*serverReference”*} [*preference*]	Adds entries to the hunt group subclassified by the option in which they are entered. Each option is a pointer to a previously configured trunk, gateway, exchange, carrier, or enterprise server. You can add multiple options of a type, for example, multiple exchange options. To add multiple types to a single option, use the pool functionality. For example,create a server configuration with a server pool that contains the servers you wish to group. Optionally, you can set a preference for the entry, which the arbiter uses in its calculations. If you do not set a preference, the value is the maximum (65535). Use the **combine-options** property to configure AA-SBC to arbitrate on all entries as if they were in a single pool. **Example:** set option server “vsp enterprise servers sip-gateway SessMgr@NYC” 5 There are no default entries. Optionally, enter a preference in the range of 0, the most preferred, to 65535. The default preference for an entry is **none** (equivalent to 65535), meaning that no preference is applied.

Property name	Description
combine-options {enabled | disabled}	*Secondary property.* Places an entire hunt-group, with all hunting options, into a single pool for arbitration. For example, it the hunt-group contains. • **option 1**—server 1, server 2, server 3 • **option 2**— server 4, server 5 Enabling the **combine-options** property forces AA-SBC to run the routing arbiter on all servers (1 through 5), placing the servers in the route set. By default (disabled), AA-SBC arbitrates the servers within each option separately. In this example, the arbiter would apply servers 1, 2, and 3 first, and place them in the route-set. AA-SBC would then run the arbiter on servers 4 and 5 and place them in the route set. **Example:** set combine-options enabled The default setting is **disabled**.
call-hunting-type {none | sequential | parallel}	*Secondary property.* Determines the order or method in which the system forwards the call to the next hop (gateway, server, or trunk). • **none**—the system forwards the call to the latest binding for the Request URI. • **sequential**— if there are two or more entries in a gateway or trunk group, the system first tries the primary and then the secondary. • **parallel**—when the system receives a call, it creates two call legs and forwards to both the primary and secondary entry. When one responds, the system disconnects the call with the other. **Example:** set call-hunting-type none The default setting is **sequential**.

class-of-service

Purpose

Creates a class of service category. This category is referenced in the trunk-group object as a basis for grouping. It is also used in the dial-plan arbiter object (**rule** property) to define the trunk-qos criteria.

Syntax

```
config vsp carriers class-of-service name
```

Properties

Property name	Description
description *string*	Assigns a text description to the class of service category. Enclose the description in quotation marks if it has spaces. **Example:** set description “Best class of service” Enter a string, in quotation marks if necessary; there is no default setting.

Carriers objects

13. CLI objects

CLI description

The CLI object allows you to set the AA-SBC command line interface display settings.

CLI object summary

The following table lists and briefly describes the **cli** object. See the following chapters for other objects in the CLI hierarchy:

- Chapter 14, “Cluster, box, and interface objects”

Object name	Description
`cli`	Configures the CLI screen settings.

cli

Purpose

Sets the CLI top-level prompt and the output display lines. Set either the maximum number of lines to display in a single screen page or set to continuous scrolling. When you use paged output, the --More-- prompt operates on the following keystrokes:

- [Enter] — Displays the next line of text
- [Tab] — Displays the remainder of the text
- [Esc], Q, or q — No more text
- Any keystroke — Displays the next page of text

To temporarily change the CLI display mode without changing the default configuration, use the display action at the top level of the CLI hierarchy.

Syntax

```
config box cli
config cluster box number cli
```

Properties

Property name	Description
prompt *text*	Sets the top-level CLI prompt. Specify an alphanumeric string up to 64 characters. Use quotation marks to enclose prompt strings that include spaces. **Example:** set prompt boston1> The default prompt is **NNOS-E>**.
banner `text`	Configures a banner that is displayed on all connected consoles. **Example:** set banner “system shutdown at noon.” There is no default setting.
display {paged integer | scrolled}	Sets the maximum number of lines to display on a CLI screen page or enables continuous page scrolling. **Example:** set display paged 20 For a paged display, specify a number of lines from 5 to 500. The default setting is **24** lines.

14. Cluster, box, and interface objects

Cluster description

The cluster object allows you to configure AA-SBC clusters. A cluster is a group of AA-SBC systems that operate within a real-time collaboration network. Systems defined in the cluster can provide synchronization and support failover operations. A AA-SBC cluster automatically distributes load among the cluster entities, and routes SIP sessions around any failed elements.

The box object allows you to configure the local box and interface settings. A local AA-SBC device is the hardware that you are managing over a local, Telnet, or Secure Shell (SSH) connection.

The interface object defines the physical interfaces—up to 20 Ethernet 1000 Mbps auto-negotiation, full-duplex interfaces (eth0 through eth 19).

cluster

Purpose

The cluster object allows you to enable media distribution between nodes within the cluster, creating media partners, using the **share-media-ports** property of this object. The pool of ports that participate in the distribution is defined using the media-ports object. From this object, you can open subobjects to first enable and configure AA-SBC devices and then Ethernet interfaces and their properties within a network cluster.

To set up load balancing across a cluster, see Configuring head-end and backing interfaces. You create backing interfaces using the sip object.

For media distribution on a system outside of the cluster as a media partner, define media partners using the partner object.

Syntax

```
config cluster
```

Properties

Property name	Description
name *text*	Sets an administrative name for this cluster. Enter up 32 alphanumeric characters. For text strings with blank spaces, enclose the strings within quotation marks (““). **Example:** set name “NN2620 cluster1 boston” There is no default setting.
share-media-ports {true | false}	Specifies whether or not to enable media distribution among the nodes within a cluster. When set to **true**, media partnering is enabled on the cluster. Media ports are defined for each interface using the media-ports object. **Example:** set share-media-ports true The default setting is **false**.
share-signaling-entries {true | false}	Specifies whether or not all boxes in a cluster exchange active SIP session information. When set to **true**, they do exchange data. If the primary link then goes down, a backup link can use SIP session information from the primary box to handle existing calls. This property should be set to **true** if you have configured VRRP (to provide the redundancy support). **Example:** set share-signaling-entries true The default setting is **false**.
backup-session-on-demand {enabled | disabled}	Specifies the manner in which failover is handled by the system. By default (disabled), if signaling failover is configured the system immediately creates a backup session on the failover box when a session is established. When this property is **enabled**, the system does not create and mirror the session on the backup box. Instead, it maintains a minimum amount of information on the backup, and establishes the full session upon failover. **Example:** set backup-session-on-demand enabled The default setting is **disabled.**

Property name	Description
share-turn-ports {true | false}	Specifies whether or not to enable distribution of TURN ALLOCATE requests (i.e., load balance) across systems within a cluster. When set to **true**, TURN partnering is enabled on the cluster (the STUN service table contains STUN/TURN routes from all systems in the cluster). AA-SBC will perform a cluster-wide STUN service route lookup to determine the best AA-SBC device to handle the ALLOCATE request. If set to **false**, the table contains only those routes for the local box. TURN servers are defined for each interface using the stun-server object. To load balance ALLOCATE requests, you must **enable** the TURN-redirector option of the stun-server **port** property for one or more STUN servers. To determine route preference for load balancing, use the stun-service-routing object. **Example:** set share-turn-ports true The default setting is **false**.
mirror-media-streams {true | false}	Specifies whether the cluster participates in media mirroring. When set to true, all boxes in the cluster share media state information. The selection of calls that are mirrored is determined by policy; you must also enable the media object **mirror** property. **Example:** set mirror-media-streams true The default setting is **false**.
media-resource-failure-timer {disabled | enabled "*integer* days *hh:mm:ss*"}	*Secondary property.* Sets how long to wait after a media proxy failure (in media proxy configurations). When **enabled**, if no backup media proxy takes ownership before the timer set with this property expires, the signaling sessions for the failed media proxy are terminated. Enter the value in the format displayed in the example or in standard W3C XML format (P*n*Y*n*M*n*DT*n*H*n*M*n*S). **Example:** set media-resource-failure-timer enabled "5 days 12:00:00" The default setting is **disabled.**

box

Purpose

Configures the settings and interfaces on the specified physical device. Specify a box number (integer) in the range 1 to 16 that makes this box unique among other boxes in the network cluster. (A box is another name for an AA-SBC blade.)

Until you configure a box, AA-SBC does not list it as an available selection when you use the help. In the following example, AA-SBC does not list box 3 as available until after you have opened the box 3 object.

```
config cluster> config box ?

 specify an object of type box

 1
 2

config cluster> config box 3
config box 3> return
config cluster> config box ?

 specify an object of type box

 1
 2
 3
```

Syntax

```
config box
config cluster box integer
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables the specified box (number) within the cluster. When disabled, you can configure box properties, but the box cannot pass traffic. **Example:** set admin enabled The default setting is **enabled**.
hostname *string*	Assigns a network hostname or IP address to the box. This name can be seen in the configuration display as well as in network traffic data. **Example:** set hostname NNOS-E-Boston Enter a string, following the *RFC 922* or *RFC 1123* hostname naming standards (a string or an IP address). There is no default setting.
timezone *zone*	Sets the time zone for the system. Use the help function to list the common values (**set timezone ?**). Or, to see the full list of supported time zones, see the file /usr/share/zoneinfo. If you enter an unsupported time zone, the system sends an error to the event log. When interpreting a time zone, the system uses the Linux standard. For time zones east of GMT, use GMT-minus-hours; for time zones west of GMT use GMT-plus-hours. For example, you would configure **timezone** as **GMT-minus-1** if GMT is one hour earlier than your local time. **Example:** set timezone GMT-plus-5 The default setting is **Eastern**.
name *string*	Assigns an administrative name for this system. Enter the name using up to 32 alphanumeric characters. For strings containing blank spaces, enclose the string within quotation marks (““). **Example:** set name bostonMaster There is no default setting.

Property name	Description
description *string*	Assigns a user-defined text description to this system. Enter the description using up to 32 alphanumeric characters. For strings containing blank spaces, enclose the string within quotation marks (““). **Example:** set description “NNOS-E network master for company HQ” The default setting is **Eclipse**.
contact *string*	Identifies the name of a contact person for this system. Enter the contact description using up to 32 alphanumeric characters. For strings containing blank spaces, enclose the string within quotation marks (““). **Example:** set contact “Bob at extension 123” There is no default setting.
location *string*	Identifies the physical location of this system. Enter the location description using up to 32 alphanumeric characters. For strings containing blank spaces, enclose the string within quotation marks (““). **Example:** set location “Data center, 2nd floor” There is no default setting.
identifier *hexValue*	Assigns a MAC address to the eth0 interface of this box. This address is used to identify the box and associate the configuration with the correct physical appliance. You only assign a MAC address to boxes in a cluster; a standalone does not require this setting. **Example:** set identifier 0f:1c:22:cd:d1:32 The default setting is **00:00:00:00:00:00**. You must change the value for the configuration to be recognized and the box to be operational.

Property name	Description
transcoding-threads *threads*	Sets the number of SIP stack processing threads that should be used for transcoding. Typically, the number of threads should match the number of system processors. **Example:** set transcoding-threads 8 Enter a value between 1 and 8. The default setting is **4** threads.
recording-socket-threads {automatic | *threads*}	Sets the number of SIP stack processing threads that should be used for servicing the recording sockets. Typically, the number of threads should match the number of system processors. (The automatic setting is based on that value.) Changes to this setting only take place after a system restart. See Using automatic values for more information. **Example:** set recording-socket-threads 4 The default setting is **automatic**.
dos-rule-source-limit *rules*	Sets the number of rules that can be evaluated strictly on source IP address. These source-only rules are evaluated before other kernel rules, and can provide faster evaluation under a heavy DOS attack. **Example:** set dos-rule-source-limit 8 Enter a value between 0 (none) and 1000. The default setting is **1000** rules.

interface

Purpose

Configures AA-SBC Ethernet network interfaces. You can configure up to 20 gigabit Ethernet, full-duplex interfaces. The actual number available depends on your hardware configuration.

It is from the interface object that you configure VLAN and IP settings:

- Chapter 77, “VLAN objects”
- Chapter 35, “IP objects”

Syntax

```
config box interface ethX
config cluster box integer interface ethX
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables the administrative state of this Ethernet interface. **Example:** set admin enabled The default setting is **enabled**.
mtu *integer*	Set the maximum transmission unit (MTU) in bytes for Ethernet packets transmitted over this interface. **Example:** set mtu 1500 Enter a value between 100 and 1500. The default setting is the maximum **1500** bytes.
arp {enabled | disabled}	Enables the Address Resolution Protocol (ARP) on this interface. ARP is the Internet protocol that maps real IP addresses to corresponding Ethernet addresses. **Example:** set arp enabled The default setting is **enabled**.
speed (10Mb | 100Mb | 1Gb}	Sets the speed of the Ethernet connection between the system and the piece of equipment to which it is connected. The system ignores this value if **autoneg** is set to enabled. **Example:** set speed 100Mb The default setting is **1Gb**.

Property name	Description
duplex {half | full}	Sets the acceptable duplex method for the interface, either half (asynchronous) or full (simultaneous) transmission. The system ignores this value if **autoneg** is set to enabled. **Example:** set duplex half The default setting is **full**.
autoneg {enabled | disabled}	Sets whether autonegotiation is enabled or disabled. If enabled, the **speed** and **duplex** settings are ignored. The system negotiates with the piece of equipment to which it is connected to achieve optimal agreed upon settings. **Example:** set autoneg disabled The default setting is **enabled**.

file-client

Purpose

Configures AA-SBC to recognize an intermediary (proxy) for file fetch operations. By specifying a match criteria for the proxy configuration, you can set AA-SBC to send different types of file requests to different proxies.

Syntax

```
config box file-client
config cluster box integer file-client
```

Properties

Property name	Description
proxy *regExp* [*host*] [*port*]	Configures the proxy that the system uses for file fetch operations. Enter a regular expression to match the target URL. Use this, for example, to send FTP requests through one proxy (match ftp://) and HTTP requests through another (match http://). To send all file requests through one proxy, match all (.*). Optionally, you can set a host name or IP address, and a port. If you set no host for a proxy match, that type will use no proxy. **Example:** set proxy ftp:// 192.168.109.10 20 There is no default setting.
http-max-redirects	The number of redirects in an HTTP URL before the AA-SBC issues a warning. **Example: set http-max-redirects 5** Min: 1/ Max: 100 The default setting is **10**.

os

Purpose

Specifies whether AA-SBC should offload incoming packets to a different CPU and the verbosity level of the information from a kernel “panic.” In addition, you can enable compression to allows at least 40% more storage for kernel dumps.

-

Note: Do not modify these values unless told to do so by Technical Support.

Syntax

```
config box os
config cluster box integer os
```

Properties

Property name	Description
rx-queue-offload {none | hyperthread-pair | alternate-cpu	Specifies whether incoming packets should be offloaded to a different CPU. Do not change this property. **Example:** set rx-queue-offload none There default setting is **none**.
crash-dump-level {disabled | header | header-kernel-page-mem | header-all-mem-except-free-pages | all-mem}	Specifies the level of information to capture when a kernel panic occurs. The dump facility captures kernel logs, kernel memory, task states, and trace information. This information is written to a file, which can later be used for debugging. The kernel header, written in all levels except disabled, contains the following information: • build time, time of the crash, kernel version, etc. • in-memory kernel logs that have not yet been written to disk. • kernel call trace details. • system task states. Select one of the following verbosity levels: • **disabled**—kernel dumping is disabled. If the kernel crashes, no extra processing occurs. • **header**—the system prints out dump information to the dump header and then writes it to the dump file. • **header-kernel-page-mem**—the system writes out the dump header all kernel memory pages to the dump file. • **header-all-mem-except-free-pages**—the system writes out the dump header and all kernel and user memory pages to the dump file. • **all-mem**—the system writes out the dump header and all conventional/cached memory (RAM) pages in the system (kernel, user, and free). Use the **show faults** command to display the name of the file the system created as a result of the kernel panic. **Example:** set crash-dump-level all-mem There default setting is **header-kernel-page-mem.**

Property name	Description
crash-dump-compression {enabled | disabled}	Specifies whether to enable GZIP compression for the dump facility. When **enabled**, the system compresses the dump image, allowing a greater number of active pages into the raw partition. This provides at least 40% more storage for kernel dumps. **Example:** set crash-dump-compression disabled The default setting is **enabled**.
arp-thresholds *threshold1 threshold2 threshold3*	Specifies the number of directly connected hosts (ARP entries in the cache) the system supports. Each of the three thresholds initiates an increasingly aggressive ARP cache action, defined as follows: • *threshold1*—specifies the point at which the system attempts to purge outdated ARP entries. Enter a value from 64 to 1,024. • *threshold2*—specifies the point at which the system aggressively tries to purge outdated ARP entries, making room for new entries. Enter a value from 128 to 4,096. • *threshold3*—specifies the maximum number of ARP entries the system supports. Any additional learned ARP entries are silently discarded until the cache entries drop below this level. Enter a value from 512 to 8,192. **Example:** set arp-thresholds 256 512 2048 There default setting for threshold 1 is **128**, for threshold 2 is **512**, for threshold 3 is **1024**.
ip-frag-queue-control	Specify the number of milliseconds the AA-SBC holds onto an IP fragment when the first fragment has not been received. The minimum valid value is 1 and the maximum valid value is 20. **Example: set ip-frag-queue-control 15** The default value is **5**.

media-partners

Purpose

Opens the object through which you identify the partners, other appliances outside of the cluster, for media distribution.

Media distribution through the **media-partners** object allows you to configure an appliance outside of the cluster as a media partner. The media partner system does not perform any SIP signaling; it has only media interfaces and handles media traffic. It offloads media anchoring to another appliance; the cluster does load balancing across its specified list of media partners.

Each cluster can specify one or more media partners. Multiple clusters can use the same media partner system(s).

For media distribution within a cluster, use the **share-media-ports** property of the cluster object.

Syntax

```
config cluster media-partners
```

Properties

None

partner

Purpose

Configures a partner, outside of the cluster, to take part in the media distribution system. See media-partners for a description of media distribution.

Enter the IP address of the partner to configure when opening this object.

➡ **Note:** The protocol, port, and certificate set within this object must match the values set for these properties in the IP messaging object. If they do not match, the systems will not be able to communicate.

Syntax

```
config cluster media-partners partner ipAddress
```

Properties

Property name	**Description**
admin {enabled | disabled}	Enables or disables the specified partner. When disabled, you can configure properties, but the partner cannot participate in shared anchoring. **Example:** set admin enabled The default setting is **enabled**.
protocol {tcp | tls}	Specifies the protocol that the partners use to communicate, either TCP or secure TLS. **Example:** set protocol tls The default setting is **TCP**.
port *portNumber*	Sets the port number to connect to on the other media partners. **Example:** set port 12112 Enter a number between 1 and 65535. The default port number is **5132**.
certificate *certificateReference*	Assigns the certificate that these media partners present if the protocol is set to TLS. Enter a reference to a previously configured certificate, used by all members of this partnership. **Example:** set certificate vsp tls certificate nnos-e.companyA.com There is no default setting.

media-anchor-limits

Purpose

Opens the object through which you set the number of active ports available for media anchoring.You can set limits on a per-box and per-card basis, preventing an individual card from becoming oversubscribed.

Syntax

```
config box media-anchor-limits
config cluster box integer media-anchor-limits
```

Properties

None

port-limit

Purpose

Sets per-processor limits on the number of ports (and thus, the number of active calls) available for media anchoring at any one time. You set limits for each processor on each “card” in the system.

You can configure AA-SBC to use ports on the **cxc** processor. Do not configure media ports on AA-SBC ports (typically eth0 through eth3).

The limits you set with this object apply to the total ports on all Ethernet interfaces for the given processor. Use the **show media-ports-process-units** command to view the limits and configuration for each processor. In the example below, across interfaces eth0 though eth3 there are a total of 10,000 media ports allocated.

```
NNOS-E> show media-ports-process-units

card process-unit busy-threshold full-limit total      active
---- ------------ -------------- ---------- -----      ------
CXC  CXC          0              16000      10000      0
```

Media ports are assigned to an IP interface via the media-ports object, which defines the starting port number and total ports available for media anchoring on a given interface. By setting port limits (this object), you are defining the total number of ports that can be simultaneously *active* for each processor on the appliance.

➡ **Note:** For purposes of calculating port requirements, typically one anchored voice call requires four ports.

To open the object, enter the card followed by the applicable processor.

Syntax

```
config box media-anchor-limits port-limit {all | CXC} {all | CXC}
config cluster box integer media-anchor-limits port-limit {all | CXC}
    {all | CXC}
```

Properties

Property name	Description
busy-threshold *integer*	Specifies the point at which the system considers the processor busy, and finds a less-congested processor, if available.This is a “soft” limit that serves to distribute anchoring across processors for best performance. For the **cxc** processor, the default is 0 so that all media anchoring will be offloaded if other processors are available. **Example:** set busy-threshold 14000 The default setting is **0** for the **cxc** processor and **12,000** for all others.
full-limit *integer*	Specifies the maximum allowable active ports for a processor. When this value is reached, no new calls can be established on the effected processor. Note that the total number of ports allocated for a processor, as displayed with the **show media-ports-processor-units** command, may be higher than this value, as those ports are not all active. The total port count may also be lower than this limit, in which case the active ports will not exceed the total. **Example:** set full-limit 18000 The default setting is **16,000**.

packet-discard

Purpose

The IP discard packet logging feature allows you to enable a discard to be logged for UDP, TCP, and “Other,” with an option to log which specific ports were hit within the previous configured scanning interval for the UDP and TCP packets.Within a given scan interval, the header of the first eight discarded packets are logged. The total discarded packets are counted and logged at the end of the scanning interval.

This feature is configured via the packet-discard object. If an IP interface has media-ports configured, you must first disable the media-ports>idle-monitor property, before the packet-discard object can be enabled. See the media-ports object for information on how to do this.

Syntax

```
config cluster box interface ip packet-discard
```

Properties

Property name	Description
admin [enabled | disabled]	Enable or disable the packet discard feature. **Example: set admin disabled** The default setting is **enabled**.
track-port [enabled | disabled]	When enabled, this will log an additional type of message with the list of ports that were hit within the logging interval. **Example: set track-port enabled** The default setting is **disabled**.
scan-interval	The interval in seconds between reading and logging the latest discarded packet information. **Example: set scan-interval 70000** Min: 10 / Max: 86400 The default setting is **60**.

lcr-import-service

Purpose

This configuration object is reserved for future use.

Syntax

```
cluster bos interface ip lcr-import-service
```

```
config cluster box integer file-client
```

Properties

Property name	Description
admin	Reserved for future use.
protocol	Reserved for future use.
max-threads	Reserved for future use.
min-spare-threads	Reserved for future use.
max-spare-threads	Reserved for future use.
idle-timeout	Reserved for future use.
ciphers	Reserved for future use.
https-call-rates-download	Reserved for future use.

15. Condition list objects

Condition list description

Condition lists help define the criteria by which decisions are made as to how to manipulate or forward calls. You can set conditions to determine policy application and to select applicable dial or registration plans (and their components).

The policy configuration object allows you to create policies that govern the routing of SIP phone calls and instant messages to recipients, and then back to the original caller or sender. A policy is identified by a unique name and can contain one or more rules. With each rule, you configure a condition list with a set of properties, (as well as the session configuration properties).

Both dial and registration plans use condition lists as a “first pass” filter when matching a plan entry. AA-SBC compares an incoming request against the configured conditions, and returns a list of matching plan entries. The match statements within the plan components then determine the final selection and/or alteration.

The following registration-plan components use condition lists:

- normalization
- arbiter
- route

The following dial-plan components use condition lists:

- normalization
- arbiter
- route
- source-route

The following calling-groups components use condition lists:

- route

- source-route

The following objects in the carriers switch and trunk-group subobjects use condition lists:

- outbound-normalization
- inbound-normalization

The following server-pool server subobjects use condition lists:

- outbound-normalization
- inbound-normalization

In addition, the condition list selection is available as a match component for URI matching that defines to which calls AA-SBC should apply the configured plan. See Using the match properties for more information.

Condition-list configuration object summary

The following table lists and briefly describes the **condition-list** object. See the following chapter for other objects in the CLI hierarchy:

- Chapter 21, “Dial plan objects”
- Chapter 46, “Outbound and inbound normalization objects”
- Chapter 48, “Policy objects”
- Chapter 55, “Registration plan objects”
- Chapter 59, “Server objects”
- Chapter 75, “Virtual system partition (VSP) objects”

Object name	Description
`condition-list`	Defines the conditions for policy and registration and dial plan matching.

condition-list

Purpose

Defines the conditions for policy, dial plan, or registration plan matching. If conditions match, the settings of the session configuration apply. Otherwise, default session configuration settings apply. Match statements in a condition-list include matching messages; exclude statements omit them. In some cases, you can specify a substring of a regular expression, which if found, will be a match. Specify a property and a regular expression or value for the corresponding variables. You must re-execute the command for each field you want included.

See Using relational operators for definitions of the operators used in condition construction.

Syntax

```
config vsp carriers carrier name gateway name inbound-normalization
   name condition-list
config vsp carriers carrier name gateway name outbound-normalization
   name condition-list
config vsp carriers carrier name gateway name trunk-group name
   inbound-normalization name condition-list
config vsp carriers carrier name gateway name trunk-group name
   outbound-normalization name condition-list
config vsp dial-plan normalization name condition-list
config vsp dial-plan arbiter name condition-list
config vsp dial-plan route name condition-list
config vsp dial-plan fork name condition-list
config vsp dial-plan voice-mail name condition-list
config vsp calling-groups group name route name condition-list
config vsp calling-groups group name source-route name condition-list
config vsp enterprise servers serverType serverName server-pool server
   serverName outbound-normalization name condition-list
config vsp enterprise servers serverType serverName server-pool server
   serverName inbound-normalization name condition-list
config vsp policies session-policies policy name rule name
   condition-list
config vsp registration-plan normalization name condition-list
config vsp registration-plan arbiter name condition-list
config vsp registration-plan route name condition-list
```

Properties

Property name	Description
operation {AND | OR}	Specifies the decision operation to use (AND/OR) should a condition match occur in the SIP call session. If multiple condition matches occur, the AND operation applies the matching rule with the highest precedence. The OR operation can use any of the matched conditions. **Example:** set operation or The default setting is **AND**.
mode {evaluate | always-true}	Sets how the system applies the condition list. If set to **evaluate**, the system runs the conditions to determine whether or not to apply the session configuration settings. If set to **always-true**, the system applies the session configuration settings, no conditions need to be configured. You may want to use this setting, for example, to apply session settings for a configured group without making up conditions to match the group. **Example:** set mode always-true The default setting is **evaluate**.
sip-message-condition	See SIP message condition options.
from-uri-condition	See From, To, and Request URI condition options.
to-uri-condition	See From, To, and Request URI condition options.
request-uri-condition	See From, To, and Request URI condition options.
date-time-condition	See Date and Time condition options.
action-condition {none | call-control | presence-subscribe | presence-end-subscription}	Specifies whether the configured rule is applied to normal SIP traffic or to a specific action. Select **none** to apply the rule to SIP traffic or select one of the supported actions. **Example:** set action-condition call-control The default setting is **none**.

SIP message condition options

Matches on general fields within the received SIP message.

Properties

Match	Description
box {eq | ne | gt | lt | ge | le} *number*	Includes or excludes messages based on the AA-SBC box number within the cluster on which this SIP message was received. **Example:** set sip-message-condition box exclude 3
vsp `{match	exclude	contains}` *regExp*	Includes or excludes messages based on the VSP on which this SIP message was received. **Example:** set sip-message-condition vsp match 1
to-uid {eq | ne | gt | lt | ge | le} *UID*	Includes or excludes messages based on the identity of the recipient of the packet. The UID is a system-generated number that assigned to all SIP users known to the system. You can view this value using the AA-SBC Management System accounting/ users screen. **Example:** set sip-message-condition to-user gt 013
from-uid {eq | ne | gt | lt | ge | le} *UID*	Includes or excludes messages based on the user identity of the sender of the packet. The UID is a system-generated number that assigned to all SIP users known to the system. You can view this value using the AA-SBC Management System accounting/ users screen. **Example:** set sip-message-condition from-user eq 025
ip-interface `{match	exclude	contains}` *regExp*	Includes or excludes messages based on the AA-SBC network interface on which the SIP message was received. **Example:** set sip-message-condition `ip-interface match 192.168*`

Match	Description
direction `{match | exclude} {RX | TX}`	Includes or excludes messages based on the direction of the packet—either TX (outbound) or RX (inbound). **Example:** set sip-message-condition `direction exclude outbound`
transport `{match | exclude} {any | UDP | TCP | TLS}`	Includes or excludes messages based on the protocol type of the packet. **Example:** set sip-message-condition `protocol exclude tls`
remote-ip `{match | exclude}` *ipAddress*	Includes or excludes messages based on the originating IP address. Enter and address and mask using CIDR notation. **Example:** set sip-message-condition `remote-ip match 10.10.0.0/24`
local-ip `{match | exclude}` *ipAddress*	Includes or excludes messages based on the destination IP address. Enter and address and mask using CIDR notation. **Example:** set sip-message-condition `local-ip match 20.20.0.0/24`
remote-port {eq | ne | gt | lt | ge | le} *port*	Includes or excludes messages based on the originating port of the packet. Enter a port number between 1 and 65535. **Example:** set sip-message-condition `remote-port eq 5040`
local-port {eq | ne | gt | lt | ge | le} *port*	Includes or excludes messages based on the AA-SBC port over which the packet is received. Enter a port number between 1 and 65535. **Example:** set sip-message-condition `local-port gt 1010`
private-remote-ip `{match | exclude}` *ipAddress/mask*	Includes or excludes messages based on the private IP address of the remote device when NAT is being used and detected by the system. **Example:** set private-remote-ip match 172.168.10.10/24

Condition list objects

Match	Description
private-remote-port {eq | ne | gt | lt | ge | le} *port*	Includes or excludes messages based on the private port of the remote device when NAT is being used and detected by the system. **Example:** set private-remote-port eq 3660
uac-public-ip `{match	exclude}` *ipAddress/mask*	Includes or excludes messages based on the public IP address of the UAC (the user agent initiating the call leg), which is present if the UAC is doing NAT. **Example:** set uac-public-ip exclude 10.10.0.0/32
uac-public-port {eq | ne | gt | lt | ge | le} *port*	Includes or excludes messages based on the public port of the UAC (the user agent initiating the call leg), which is present if the UAC is doing NAT. **Example:** set uac-public-port eq 10.10.0.0/32
`call-leg {match	exclude} {inbound	outbound}`	Includes or excludes messages based on the direction of the packet—inbound or outbound. **Example:** set sip-message-condition `call-leg exclude outbound`
`message-type {match	exclude} {request	response}`	Includes or excludes table rows based on the type of message—request or response. **Example:** set sip-message-condition `message-type match response`

Match	Description
`request-method {match | exclude}` *`method`*	Includes or excludes messages based on their request method. Select a standard method: • INVITE • ACK • OPTIONS • BYE • CANCEL • REGISTER • MESSAGE • INFO • NOTIFY • SUBSCRIBE • REFER • PRACK • PUBLISH • UPDATE **Example:** set sip-message-condition `request-method exclude info`
request-uri `{match | exclude | contains}` *`regExp`*	Includes or excludes table rows based on the string found in the request URI field. **Example:** set sip-message-condition `request-uri match *badguy*`

`response-code {eq | ne | gt | lt | ge | le}` *`code`*	Includes or excludes messages based on the value in the response code field. Enter a code number or select a standard code. (Enter a ? after the mathematical operator to see the list of standard codes. For example, **set sip-message-condition response-code eq ?**) You can use the mathematical operators to specify a range of codes.
	Example: **`set sip-message-condition`**
	`response-code match ge 600`
	`set sip-message-condition`
	`match le 606`

response-string `{match | exclude | contains}` *`regExp`*	Includes or excludes messages based on the string found in the response field of the SIP header. **Example:** set sip-message-condition `response-string exclude OK`

Condition list objects

Match	Description
call-id `{match | exclude | contains}` *`regExp`*	Includes or excludes messages based on the value in the call ID field. **Example:** set sip-message-condition `call-id match 3ab7d43aab0d43dbbec041a*`
from `{match | exclude | contains}` *`regExp`*	Includes or excludes messages based on the value in the From: field of the SIP header. **Example:** set sip-message-condition `from exclude mis@companyABC.com`
to `{match | exclude | contains}` *`regExp`*	Includes or excludes messages based on the value in the To: field of the SIP header. **Example:** set sip-message-condition `to match *.eng.*@companyABC.com`
cseq `{match | exclude | contains}` *`regExp`*	Includes or excludes messages based on the value in the command sequence field. Use quotation marks to enclose strings that include spaces. **Example:** set sip-message-condition `cseq match “3 invite”`
`content-type {match | exclude | contains}` *`regExp`*	Includes or excludes messages based on the string in the content type field of the SIP header. **Example:** set sip-message-condition `content-type match application/*`
user-agent `{match | exclude | contains}` *`regExp`*	Includes or excludes messages based on the specified string being found in the SIP header field of the packet. **Example:** set sip-message-condition `user-agent exclude *.win.*`
`header {match | exclude | contains}` *`regExp`*	Includes or excludes messages based on the specified string being found in the SIP header field of the packet. **Example:** set sip-message-condition `header match .*boom.*`

Match	Description
`content {match | exclude | contains}` *`regExp`*	Includes or excludes messages based on the specified string found in the packet payload. **Example:** set sip-message-condition `content match “What are you doing Dave?”`
result-code `{match | exclude} {success | bad-header | policy-discard | policy-refuse | socket-timeout}`	Includes or excludes messages based on the value of the result field. Select one of the following, which indicates: • **success**—the packet went through. • **bad-header**—the system could not parse the header and so discarded the packet. • **policy-discard**—the system immediately discarded the packet due to a session policy firing with an action set to discard. • **policy-refuse**—the system discarded the packet, due to a session policy firing with an action set to discard, but sent a response to indicate having done so. • **socket-timeout**—the packet was dropped due to a socket timeout. **Example:** set sip-message-condition `result exclude success`
result-description `{match | exclude | contains}` *`regExp`*	Includes or excludes messages based on any further information they may have been included in the results (such as a descriptive string for the resultCode). **Example:** set sip-message-condition `result-string match “from badguy.policy”`
uac-public-transport {match | exclude} {any | UDP |TCP | TLS}	Includes or excludes messages based on the transport protocol used by the UAC (the user agent initiating the call leg), which is present if the UAC is doing NAT. **Example:** set uac-public-transport match TLS

Condition list objects

Match	Description
request-uri-user-host `{match | exclude | contains}` *`regExp`*	Includes or excludes messages based on the “user@host” portion of the Request URI. **Example:** set request-uri-user-host match exec@company.com
to-user-host `{match | exclude | contains}` *`regExp`*	Includes or excludes messages based on the “user@host” portion of the TO URI. **Example:** set to-user-host match admin@company.com
from-user-host `{match | exclude | contains}` *`regExp`*	Includes or excludes messages based on the “user@host” portion of the FROM URI. **Example:** set to-user-host exclude joe@company.com

Match	Description
cseq-message-type `{match | exclude}` *`methodType`*	Includes or excludes messages based on the SIP method type from the CSeq: header line. Select one of the following method types: • INVITE • ACK • BYE • REGISTER • REFER • NOTIFY • OTHER • PRACK • CANCEL • SUBSCRIBE • OPTIONS • MESSAGE • INFO • PUBLISH • UPDATE • SERVICE • NONE **Example:** set sip-message-condition `cseq-message-type exclude INFO` The default setting is **match** and **INVITE**.
media-types `{match | exclude | contains}` *regExp*	Includes or excludes messages based on the media (CODEC) type. If you set this attribute, you must enable the **prescan-media-types** property of the VSP settings object. Note that this attribute is case-insensitive. **Example:** set media-types contains g726

From, To, and Request URI condition options

Includes or excludes messages based on the content of the To, From, or Request URI field strings in the SIP message.

Properties

Match	Description
`scheme {match | exclude}` *regExp*	Includes or excludes messages based on the URI type: SIP, SIPS, or Tel. **Example:** set from-uri-condition `scheme match sips`
`user {match | exclude}` *`regExp`*	Includes or excludes messages based on the user portion of the To, From, or Request URI. **Example:** set from-uri-condition `user exclude *bob.roberts*`
`user-param {match | exclude}` *`regExp`*	Includes or excludes messages based on a user parameter in the SIP header of the To, From, or Request URI. **Example:** set from-uri-condition `user-param exclude *=internal`
`host {match | exclude}` *`regExp`*	Includes or excludes messages based on the SIP server host portion of the To, From, or Request URI. **Example:** set from-uri-condition `host exclude *cov.com*`
`port {match | exclude}` *`regExp`*	Includes or excludes messages based on the port number recorded in the To, From, or Request URI. **Example:** set from-uri-condition `port match 5060`
`ttl {match | exclude}` *`regExp`*	Includes or excludes messages based on the time-to-live value associated with the UDP multicast packet for the particular URI type. **Example:** set from-uri-condition `ttl exclude 15`
`method {match | exclude}` *`regExp`*	Includes or excludes messages based on the SIP request method: INVITE, REGISTER, NOTIFY, etc. **Example:** set from-uri-condition `method match INVITE`

Match	Description
`url {match	exclude}` *`regExp`*	Includes or excludes messages based on the SIP URL. **Example:** set from-uri-condition `url match *companyABC*.com`
`other {match	exclude}` *`regExp`*	Includes or excludes messages based on any parameters in the SIP header of the To, From, or Request URI *except* the user parameter. **Example:** set from-uri-condition `other exclude match *=temp`
`transport {match	exclude} {any	UDP	TCP	TLS}`	Includes or excludes messages based on the transport protocol type of the packet: UDP, TCP, TLS, or any protocol. **Example:** set from-uri-condition `protocol exclude tls`

Date and Time condition options

Creates a match condition based on the date and time of the SIP message. You can enter as many date time conditions as you wish. Use multiple conditions to further refine the date/time.

Properties

Match	Description
`hour {eq | ne | gt | lt | ge | le}` *`value`*	Specifies the hour in the time string of the SIP message to match on for this rule to apply. Enter a whole number between 0 (midnight) and 23. To create times that do not fall on the hour boundary, for example, 1:30, use the hour and minute conditions **Example:** **`set date-time-condition hour gt 5`** **`set date-time-condition hour lt 21`**
`minute {eq | ne | gt | lt | ge | le}` *`value`*	Specifies the minute in the time string of the SIP message to match on for this rule to apply. Enter a whole number between 0 and 59. **Example:** set date-time-condition `minute eq 30`
`day {match | exclude} {Sunday | Monday | Tuesday | Wednesday | Thursday | Friday | Saturday}`	Specifies the day of the week in the time string of the SIP message to match on for this rule to apply. **Example:** **`set date-time-condition exclude Saturday`** **`set date-time-condition exclude Sunday`**
`date {eq | ne | gt | lt | ge | le}` *`value`*	Specifies the date in the time string of the SIP message to match on for this rule to apply. Enter a whole number between 1 and 31. **Example:** set date-time-condition `date gt 15`
`month {match | exclude}` *`month`*	Specifies the month in the time string of the SIP message to match on for this rule to apply. Enter the name of a month. **Example:** set date-time-condition `month exclude January`
`year {eq | ne | gt | lt | ge | le}` *`value`*	Specifies the year in the time string of the SIP message to match on for this rule to apply. Enter any value. **Example:** set date-time-condition year eq 2006

16. Console objects

Console description

The console object configures the serial port and remote console settings for a box.

Console object summary

The following table lists and briefly describes the **console** objects and properties. See the following chapters for other objects in the CLI hierarchy:

- Chapter 14, “Cluster, box, and interface objects”

Object name	Description
console	Opens the console configuration object for editing.
remote	Configures remote console access.

console

Purpose

Configures the serial port settings for the AA-SBC device.

Syntax

```
config box console
config cluster box number console
```

Properties

-

Property name	Description
rate *rate*	Sets the baud rate in bits per second. Enter **set rate ?** at the prompt to see available rates. **Example:** set rate 460800 The default setting is **115200**.
data-bits {5 | 6 | 7 | 8}	Sets the number of data bits used. **Example:** set data-bits 6 Select either 5, 6, 7, or 8. The default setting is **8**.
parity {none | even | odd}	Sets the parity value used for parity checking. Select either **none** (no parity checking), **even**, or **odd**. **Example:** set parity even The default setting is **none**.
stop-bits {1 | 2}	Sets the number of stop-bits used for data sent from the box. **Example:** set stop-bits 2 Select either 1 or 2. The default setting is **1**.
flow-control {none | xon-xoff}	Sets the flow control mechanism to be used on the box. Select either **none** or **xon-xoff**. **Example:** set flow-control xon-xoff The default setting is **none**.

remote

Purpose

Configures remote console access for the system via the eth0 interface. When configured, a client using the Serial Over LAN protocol can connect to the remote console, providing access similar to a direct serial port connection.

Be cautious when configuring this object. When enabled, this object makes the serial console accessible. If eth0 is configured as part of a private network and is behind a firewall, remote console access is secure. However, if the interface is public, enabling this object makes the serial console publicly available. Do not configure this object without first contacting Technical Support.

Syntax

```
config box console remote
config cluster box number console remote
```

Properties

Property name	**Description**
admin {enabled | disabled}	Sets the administrative state of the remote console configuration. **Example:** set admin disabled The default setting is **enabled**.
ip-address *ipAddress/mask*	Sets the IP address, on the eth0 interface, for the remote console. **Example:** set ip-address 172.26.0.30/32 There is no default setting (0.0.0.0\0).
default-gateway *ipAddress*	Sets the IP address for the default gateway used to reach this console address. **Example:** set default-gateway 172.26.0.1 There is no default setting (0.0.0.0).

Property name	Description
username *string*	Configures the username required for console access. The system will prompt users for this name when they attempt to access the remote console. **Example:** set username admin There is no default setting.
password-tag *string*	Configures the password required for remote console access. The system will prompt users for this password when they attempt to access the remote console. See Understanding passwords and tags for information on the two-part password mechanism. **Example:** set password-tag root There is no default setting.

17. Database object

Database description

The **database** object defines, for each history table, the number of days of records the AA-SBC device should keep when it runs maintenance operations. Use the master-service database object to schedule maintenance.

Database object summary

The following table lists and briefly describes the **database** objects.

Object name	Description
`database`	Configures the duration of table records saved when AA-SBC runs maintenance.

database

Purpose

Sets the number of days of records and archives that should be saved when AA-SBC performs database maintenance. Maintenance intervals are set using the **database** object in master services. Maintenance defines the time for or frequency of database purging. When the purge occurs, AA-SBC then keeps the number of days of table history for each table counting backward from that time. For example, if maintenance is set to occur every 96 hours and media history is saved for seven days, every 96 hours AA-SBC removes all records older than 7 days from the media message table.

➡ **Note:** The database-write property must be enabled in the vsp object for AA-SBC to successfully write data to the database.

Note on aging out entries

The database is comprised a variety of tables. The **history** properties define how long entries can stay in the associated table. The following table lists the properties and their associated tables.

Property	Associated table
accounting-history	Accounting call structure table
call-details-history	Transport message, SIP message, and session tables
media-history	Media message table
file-transfer-history	File transfer message table
im-history	Archive IM message table

Note that if any of the media, file transfer, or IM history tables have an entry for a particular sessionID, then AA-SBC will not delete that entry in the Session table, even if it is older than the call-details-history boundary. Once that sessionID has been completely aged out of all three tables, AA-SBC will then remove it from the Session table.

Database object

Syntax

```
config vsp database
```

Properties

Property name	Description
accounting-history *days*	Defines the number of days of accounting history the system should preserve when purging the accounting table. See Note on aging out entries for additional information. **Example:** set accounting-history 500 Enter a value between 1 and 1,000,000. The default setting is **365** days.
call-details-history *days*	Defines the number of days of entries the system should preserve when purging the Transport message, SIP message, and AA-SBC session tables. See Note on aging out entries for additional information. **Example:** set call-details-history 5 Enter a value between 1 and 1,000,000. The default setting is **3** days.
media-history *days*	Defines the number of days of entries the system should preserve when purging the sessions containing audio recordings. See Note on aging out entries for additional information. **Example:** set media-history 5 Enter a value between 1 and 1,000,000. The default setting is **7** days.

Property name	Description
file-transfer-history *days*	Defines the number of days of entries the system should preserve when purging the sessions containing file transfers. See Note on aging out entries for additional information. **Example:** set file-transfer-history 5 Enter a value between 1 and 1,000,000. The default setting is **7** days.
im-history *days*	Defines the number of days of entries the system should preserve when purging the IM archive tables. See Note on aging out entries for additional information. **Example:** set im-history 180 Enter a value between 1 and 1,000,000. The default setting is **365** days.

18. Default session configuration objects

Default session configuration description

The default session configuration defines the policy settings to apply to those SIP calls for which there are no specific configured policies. When a SIP call is received, AA-SBC registers the call and checks all policies and rules to determine how the call should be processed, including those services (such as authentication and accounting services) that should be applied to the SIP call.

If there are no policies that specifically match the SIP call registration information, the call is either forwarded to the SIP call recipient or the call is dropped based on the settings in the default session configuration.

For a description of all default session configuration subobjects, see Chapter 62, “Session configuration objects”.

For more information on AA-SBC policies, refer to the *Net-Net OS-E – Session Services Configuration Guide*.

Default-session-config object summary

The following table lists and briefly describes the **default-session-config object**. See the following chapter for other objects in the CLI hierarchy:

- Chapter 62, “Session configuration objects”
- Chapter 75, “Virtual system partition (VSP) objects”

Object name	Description
`default-session-config`	Opens the default session configuration object for editing. For descriptions of all session configuration subobjects, see Chapter 62, “Session configuration objects”

default-session-config

Purpose

Opens the default session configuration object for editing, which is where you specify the default configuration used for all sessions before policy is applied. For a description of all default session configuration subobjects, see Chapter 62, “Session configuration objects”.

Syntax

```
config vsp default-session-config
```

Properties

None

dialog-control-settings

Purpose

Allows you to configure the AA-SBC to reject a message sent within a dialog that contains specified release code and text..

Syntax

```
config vsp default-session-config dialog-control-settings
```

Properties

Property name	Description
refused-methods <source> <code> <text>	Select the type of message you want the AA-SBC to reject. The following are valid values: • INVITE • ACK • BYE • REGISTER • REFER • NOTIFY • OTHER • PRACK • CANCEL • SUBSCRIBE • OPTIONS • MESSAGE • INFO • PUBLISH • UPDATE • SERVICE • PING • NONE Specify the release code you want the AA-SBC to reject. The minimum value is 400 and the maximum value is 499. The default value is 405. Specify the text you want the AA-SBC to reject. The default setting is Method Not Allowed. **Example: `set refused-methods invite 450 Method Rejected`**

Default session configuration objects

19. Denial of Service (DOS) objects

The policies configuration object sets both operating policy for enterprise servers and users and denial of service (DOS) policy. This chapter details DOS policy. For information on vsp, server, and user policy, see Chapter 48, “Policy objects”.

DOS policy configuration description

Denial of service (DOS) attacks are designed to disable networks by flooding them with useless traffic. AA-SBC provides transport-layer and SIP-layer query and policy capabilities to manage DOS attacks. Queries allow you to sort and view incoming and outgoing traffic in an effort to better define policies. You can use policies to determine if a packet is attacking the box, and configure the responding action. These tools quickly identify and shutout dubious traffic, thereby limiting the damage caused by DOS attacks.

AA-SBC database records all packets that are transmitted or received by the system in specific tables that AA-SBC can then access for queries and to apply policy. Activities that happen through the transport layer, such as file transfers and SNMP walks, are stored in the transport layer table. The SIP table contains entries for all SIP-related activities.

The settings of the condition list in the transport and SIP policies determine the point at which activity is determined to be part of a DOS attack and what action is taken. The action taken by AA-SBC depends on whether the attack was identified by the transport policy or the SIP policy. A policy fires at the frequency defined by the period property, scanning the database over the course of the last period looking for matches to the policy.

AA-SBC automatically creates DOS rules as a result of the configured policies. The policy selects which table rows to consider, the threshold for instances, and the frequency (period) of comparison. When the threshold is reached for a given period, AA-SBC generates a rule in the kernel to block traffic meeting that selection criteria. The rule will persist as long as any traffic matching the rule is seen within the user-configurable inactivity-timeout period. AA-SBC automatically deletes the rule when the inactivity timer expires, or you can delete all rules manually using the **dos-delete-rules** action.

Note: AA-SBC accepts either a well-known name or number in all the port fields. For example, port name could be either sip-tls or any number, such as 5066.

For more complete information on DOS policy operations, see *Net-Net OS-E – Session Services Configuration Guide*.

Managing DOS policy results

There are several mechanisms for observing the effectiveness of your DOS policy configuration:

- The system generates an SNMP trap and a log message each time a DOS policy detects a DOS attack.
- The **show dos-sip-summary** and **show dos-transport-summary** commands display a summary of relevant traffic, detailing a count of each sending (remote IP) address, port, and destination address.
- The **show dos-sip-counters**, **show dos-transport-counters** and **show dos-url-counters** commands display a running hit count total for each policy (not rule) since the last system boot.
- The **show dos-rules** command displays a policy summary.
- The **show dos-recent-***identifier* commands display per-port and per-IP address policy hit results for either SIP or transport policies.
- The **trace dos** commands set traces on DOS activity, monitored by severity level.
- The **dos-delete-rules** action removes all automatically generated rules from the policy.

- The DOS engine generates SNMP traps when it determines there has been a qualified attack (DosTransportPolicyTrap, DosSIPPolicyTrap, DosURLPolicyTrap).

DOS object summary

The following table lists and briefly describes the `dos` objects. See the following chapter for other objects in the CLI hierarchy:

- Chapter 75, “Virtual system partition (VSP) objects”

Object name	Description
`policies`	Opens the policy configuration object.
`dos-policies`	Opens the DOS configuration object, allowing you to configure transport- and SIP-layer policy filters.
`transport-policy`	Configures a transport layer DOS policy.
`transport-condition-list`	Specifies a list of conditions for transport layer DOS policies, defining activity considered to be a DOS attack and the action taken.
`sip-policy`	Configures a SIP-layer DOS policy.
`sip-condition-list`	Specifies a list of conditions for SIP layer DOS policies, defining activity considered to be a DOS attack and the action taken.
`url-policy`	Configures a URL-based DOS policy.
`url-condition-list`	Specifies a list of conditions for URL-based DOS policies, defining activity considered to be a DOS attack and the action taken.

policies

Purpose

Opens the policies object for editing. It is through this object that you configure DOS policy and VSP-based user and server policy. For information on vsp, server, and user policy, see Chapter 48, “Policy objects”.

Syntax

```
config vsp policies
```

Properties

None

dos-policies

Purpose

Opens the DOS policies object. It is through this object that you create transport, SIP, and URL policy condition lists, which define the point at which activity is determined to be part of a DOS attack and what action is taken.

Syntax

```
config vsp policies dos-policies
```

Properties

None

transport-policy

Purpose

Opens the transport policy object. It is through this object that you define the condition list to apply to entries in the transport table. When the criteria defined in this object match incoming packets, AA-SBC kernel rules define the action to take.

The properties you set in the transport policy object define the “rules” for applying the condition list to the transport table. Some of the criteria include specifying the elements of the transport table to be considered, the number of questionable packets allowed, the interval at which AA-SBC checks the table, and the length of time a sender must be “clean” before the system restores access.

At the transport level, AA-SBC always filters based on the remote IP address field of the TCP header. In addition, it can also filter using the IP header fields:

- Remote port
- Local port
- Protocol

Note that the value of these fields does not need to be set. You are not setting the value of a field to match on. Instead, you are configuring AA-SBC to collect data from specific table columns and count occurrences of repeated values, whatever they may be, across all selected fields.

When opening the transport policy object, you must supply a name to identify the policy. Entering an existing name opens that policy for editing; entering a new name creates a new transport policy by that name.

Syntax

```
config vsp policies dos-policies transport-policy name
```

Properties

Property name	Description
description *text*	Associates a description with the named policy. That description is then displayed when you display the configuration. Using a description allows you to attach information, such as the reason for creating the policy, without giving the policy a name that is burdensome to type each time you want to configure the rule. Enter a string up to 32 alphanumeric characters; use quotation marks if there are spaces in the description. **Example: set description “Filter out periodic blastings”** There is no default setting.
admin {enabled | disabled}	Enables or disables this transport policy. **Example: set admin enabled** The default setting is **enabled**.
select {remote-port | local-port | protocol}	Sets the fields from the TCP header (and therefore, the rows of the transport table) to aggregate and compare against the condition list. The remote (originating) IP address of the packet is always part of the compare. In addition to checking the remote IP, you can optionally enter one or more of the following fields, separated by a plus (+) sign: • **remote-port**—Compares the originating port of the packet • **local-port**—Compares destination port of the packet • **protocol**—Compares the protocol type of the packet **Example: set select remote-port+ protocol** The default setting is to compare against the remote IP field.

Property name	Description
condition-list *conditionListReference*	References a previously configured condition list. The list sets the rules that apply to the columns from your transport table. **Example: set condition-list vsp policies dos-policies transport-condition-list 2** There is no default condition list setting.
threshold *integer*	Sets the number of unique instances (incoming packets to the system) allowed before the traffic is considered a DOS attack. This is the number of packets matching this policy (the conditions set forth in the list) that are allowed through before the system creates a dynamic rule in the kernel filter to block packets matching the identified pattern. **Example: set threshold 50** Enter a value between 20 and 1,000,000. The default setting is **1000** instances.
period *seconds*	Configures the interval at which you apply the transport policy to the transport table database. As a result, it also defines how many seconds of data to analyze. For example, if the period is set to 45, the system will scan the transport table every 45 seconds, and look at the last 45 seconds worth of entries. **Example: set period 45** Enter a value between 30 and 3600. The default setting is **30** seconds.

Property name	Description
remote-ip-netmask *bits*	*Secondary property.* Specifies a subnet address, allowing the system to detect and prevent DOS attacks emanating from a subnet of addresses. For example, if you set the mask to /24, the system looks for “too many instances” of packets coming from x.y.z.0/24. This is useful because an attacker may be able to spoof a source IP address, but may be prevented from spoofing outside of the range of IP addresses owned by the ISP. **Example: set remote-ip-mask 24** Enter a value from 16 to 32; the default setting is a **32-bit** mask.
inactivity-period *time*	*Secondary property.* Defines the period of time a policy must remain inactive before its effects are removed. This is the point when packets that were being denied due to a previous policy action can again be accepted. See Setting time and time intervals for information on entry format requirements. **Example: set inactivity-timeout 600** Enter a value between 30 seconds and 30 days. The default setting is **300** seconds (5 minutes).

transport-condition-list

Purpose

Opens the transport policy condition list object. It is through this list that you set the rules that apply to the columns from your transport table (aggregated using the **select** property of the **transport-policy** object). The condition list sets the criteria for choosing which packets then get run through the **select** screening to build the final result set.

See Using relational operators for information on the meaning of the property qualifiers.

The following table lists the well-known ports AA-SBC accepts for the **local-port** and **remote-port** conditions:

Well-known name	Port number	Definition
ftp-data	20	file transfer default data
ftp	21	file transfer control
ssh	22	SSH Remote Login Protocol
telnet	23	Telnet
nameserver	42	host name server
domain	53	domain name server
bootps	67	Bootstrap Protocol server
bootpc	68	Bootstrap Protocol client
tftp	69	Trivial File Transfer
www	80	World Wide Web HTTP
kerberos	88	Kerberos
ntp	123	Network Time Protocol
snmp	161	SNMP
ldap	389	Lightweight Directory Access Protocol
https	443	HTTP over TLS/SSL
syslog	514	Syslog
radius	1812	RADIUS
radius-acct	1813	RADIUS accounting

Well-known name	Port number	Definition
nat-stun-port	3478	Simple Traversal of UDP Through NAT (STUN)
diameter	3868	Diameter protocol
sip	5060	SIP
sip-tls	5061	SIP-TLS

Syntax

```
config vsp policies dos-policies transport-condition-list name
```

Properties

Property name	Description
description *text*	Associates a description with the named condition list. Enter a string up to 32 alphanumeric characters; use quotation marks if there are spaces in the description. **Example: set description “List for IP 1.1.1.1”** There is no default setting.
operation {AND | OR}	Specifies whether the conditions (properties) selected are AND'd or OR'd together. **Example: set operation OR** The default setting is **AND**.
condition *property*	Specifies the set of rows to include or exclude from this policy. Match statements include those rows; exclude statements omit them. Specify a property (all properties listed below) and the corresponding variables. You must re-execute the command for each field you want included. For properties that use a match or exclude statement, the default is match. For properties that use numeric comparisons, the default is eq (equals).
`remote-ip {match	exclude}` *ipAddress/mask*	Includes or excludes table rows based on the originating IP address. Enter an address and mask using CIDR notation. **Example: set condition `remote-ip match 10.10.0.0/16`**

Property name	Description
`remote-port` {eq | ne | gt | lt | ge | le} *port*	Includes or excludes table rows based on the originating port of the packet.Enter a port number or choose a well-known port by name. If you use a well-known port name, the system does a comparison against the known port number. **Example: set condition `remote-port eq ftp`**
`local-port` {eq | ne | gt | lt | ge | le} *port*	Includes or excludes table rows based on the port over which the packet is received. Enter a port number or choose a well-known port by name. If you use a well-known port name, the system does a comparison against the known port number. **Example: set condition `local-port eq sip-tls`**
`protocol {match | exclude} {all | icmp | tcp | udp | vrrp}`	Includes or excludes table rows based on the protocol type of the packet. **Example: set condition `protocol exclude icmp`**

sip-policy

Purpose

Opens the SIP policy object. It is through this object that you define the condition list to apply to entries in the SIP table. At the SIP level, AA-SBC can filter on data based on fields of the SIP header. There are many fields to choose from; you must define your aggregated fields with the **select** property. Note that the value of these fields does not need to be set. You are not setting the value of a field to match on. Instead, you are configuring AA-SBC to collect data from specific table columns and count occurrences of repeated values, whatever they may be, across all selected fields.

The properties you set in the SIP policy object define the “rules” for applying the condition list to the SIP table. Some of the criteria include specifying the elements of the SIP table to be considered, the number of questionable packets allowed, the interval at which AA-SBC checks the table, and an inactivity timer for the policy.

Finally, you can set an action to take on existing calls when a criteria “hit” puts a policy in place.

When opening the SIP policy object, you must supply a name to identify the policy. Entering an existing name opens that policy for editing; entering a new name creates a new transport policy by that name.

Syntax

```
config vsp policies dos-policies sip-policy name
```

Properties

Property name	Description
description *string*	Associates a description with the named policy. That description is then displayed when you display the configuration. Using a description allows you to attach information, such as the reason for creating the policy, without giving the policy a name that is burdensome to type each time you want to configure the rule. Enter a string; use quotation marks if there are spaces in the description. **Example: set description "Filter out INVITEs from 192.168.10.10"** There is no default setting.
admin {enabled | disabled}	Enables or disables this SIP policy. **Example: set admin enabled** The default setting is **enabled**.

Property name	Description
select *field*	Sets the properties to aggregate and compare against the condition list. Properties include fields from the SIP, TCP, or UDP header and internal enumerations. Enter one or more of the following, separated by a (+) sign: • **to-user**—Compares the identity of the recipient of the packet. • **from-user**—Compares the identity of the sender of the packet. • **protocol**—Compares the protocol type of the packet. • **remote-ip**—Compares the originating IP address of the packet. • **remote-port**—Compares the originating port of the packet. • **local-port**—Compares the port over which the packet is received. • **message-type**—Compares packets by their type, either request or response. • **request-method**—Compares packets according to their request method. • **request-uri**— Compares packets according to their request URI field. • **response-code**—Compares packets according to their response code fields. • **response-string**—Compares packets according to their response string field. • **call-id**—Compares packets according to their call ID field, which identifies a particular invitation and all subsequent related transactions. • **c-seq**—Compares packets according to their command sequence value in the cSeq field. • **content-type**—Compares packets according to their content type field. • **user-agent**—Compares packets according to their user agent. **Example: set select from-user+ remote-ip+remote-port** The default setting is **from-user**.

Property name	**Description**
condition-list *conditionListReference*	References a previously configured condition list. The list sets the rules that apply to the columns from your SIP table. **Example: set condition-list vsp policies dos-policies sip-condition-list 2** There is no default condition list setting.
threshold *integer*	Sets the number of unique instances of aggregated selected fields (from incoming packets to the system) allowed before the traffic is considered a DOS attack. This is the number of packets matching this policy (the conditions set forth in the list) that are allowed through before the system takes the configured action. **Example: set threshold 50** Enter a value between 20 and 1,000,000. The default setting is **60** instances.
period *seconds*	Configures the interval at which you apply the SIP policy to the SIP table database. As a result, it also defines how many seconds of data to analyze. For example, if the period is set to 45, the system will scan the SIP table every 45 seconds, and look at the last 45 seconds worth of entries. **Example: set period 45** Enter a value between 30 and 3600. The default setting is **30** seconds.

Property name	Description
inactivity-period *time*	*Secondary property.* Defines the period of time a policy must remain inactive before its effects are removed. This is the point when packets that were being denied due to a previous policy action can again be accepted. See Setting time and time intervals for information on entry format requirements. **Example: set inactivity-timeout 600** Enter a value between 30 seconds and 30 days. The default setting is **300** seconds (5 minutes).
action {filter | alert}	*Secondary property.* Sets the action taken by the system on existing calls when a dynamic policy rule is instituted in response to the threshold being crossed. Select one of the following actions: • **filter**—blocks any packet that the SIP DoS policy detected (that matches the rule defined in the condition list). When this happens, the system generates an SNMP trap and a syslog message. • **alert**—the system generates an SNMP trap and a syslog message but does not block the precipitating traffic. **Example: set action alert** The default action is **filter**.

sip-condition-list

Purpose

Opens the condition list object. It is through this list that you set the rules that apply to the columns from your SIP table (aggregated using the **select** property of the **sip-policy** object). The condition list sets the criteria for choosing which packets then get run through the **select** screening to build the final result set.

See Using relational operators for information on the meaning of the property qualifiers.

Syntax

```
config vsp policies dos-policies sip-condition-list name
```

Properties

Property name	Description
description *text*	Associates a description with the named condition list. Enter a string up to 32 alphanumeric characters; use quotation marks if there are spaces in the description. **Example: set description "Exclude user bad-guy"** There is no default setting.
operation {AND | OR}	Specifies whether the conditions (properties) selected are AND'd or OR'd together. **Example: set operation OR** The default setting is **AND**.
condition *property*	Specifies the set of rows to include or exclude from this policy. Match statements include those rows; exclude statements omit them. Specify a property (all properties listed below) and the corresponding variables. You must re-execute the command for each field you want included. For properties that use a match or exclude statement, the default is match. For properties that use numeric comparisons, the default is eq (equals).
`to-user {match | exclude}` *name*	Includes or excludes table rows based on the identity of the recipient of the packet. Enter a user name. **Example: set condition to-user exclude SIP:outfield@bball.com**
`from-user {match | exclude}` *name*	Includes or excludes table rows based on the identity of the sender of the packet. Enter a user name. **Example: set condition from-user exclude SIP:infield@bball.com**
`protocol {match | exclude} {any | UDP | TCP | TLS}`	Includes or excludes table rows based on the protocol type of the packet. **Example: set condition `protocol exclude tls`**

Property name	Description
`remote-ip {match | exclude}` *ipAddress/mask*	Includes or excludes table rows based on the originating IP address. Enter an address and mask using CIDR notation. **Example: set condition `remote-ip match 10.10.0.0/16`**
`remote-port` {eq | ne | gt | lt | ge | le} *port*	Includes or excludes table rows based on the originating port of the packet. Enter a port number or select a well-known port (sip or sip-tls). **Example: set condition `remote-port eq sip-tls`**
`local-port` {eq | ne | gt | lt | ge | le} *port*	Includes or excludes table rows based on the port over which the packet is received. Enter a port number or select a well-known port (sip or sip-tls). **Example: set condition `local-port gt 1010`**
`message-type {match | exclude} {request | response}`	Includes or excludes table rows based on the type of message—request or response. **Example: set condition `message-type match response`**
`request-method {match | exclude}` *`string`*	Includes or excludes table rows based on their request method. Enter a method type, or select a standard method. **Example: set condition `request-method exclude info`**
`request-uri {match | exclude}` *`regExp`*	Includes or excludes table rows based on the string found in the request URI field. **Example: set condition `request-uri match *badguy*`**

Property name	Description
property: `response-code {eq | ne | gt | lt | ge | le}` *`code`*	Includes or excludes table rows based on the value in the response code field. Enter a code number or select a standard code. (Enter a ? after the mathematical operator to see the list of standard codes. For example, **set condition response-code eq ?**) You can use the mathematical operators to specify a range of codes. **Example: `set condition response-code match ge 600`** **`set condition response-code match le 606`**
property: `response-string {match | exclude}` *`regExp`*	Includes or excludes table rows based on the string found in the response field. **Example: set condition `response-string exclude OK`**
property: `call-id {match | exclude}` *`regExp`*	Includes or excludes table rows based on the value in the call ID field. **Example: set condition `call-id match 3ab7d43aab0d43dbbec041a*`**
property: `c-seq {match | exclude}` *`regExp`*	Includes or excludes table rows based on the value in the command sequence field. Use quotation marks to enclose strings that include spaces. **Example: set condition `c-seq match “3 invite”`**
property: `content-type {match | exclude}` *`regExp`*	Includes or excludes table rows based on the string in the content type field of the SIP header. **Example: set condition `content-type match application/*`**
property: `user-agent {match | exclude}` *`regExp`*	Includes or excludes table rows based on the string in the user agent field of the SIP header, which describes the type of equipment originating the call. **Example: set condition `user-agent match .*RTC`**

Property name	Description
property: `header {match | exclude}` *`regExp`*	Includes or excludes table rows based on the specified string being found in the SIP header field of the packet. **Example: set condition `header match .*boom.*`**
property: `content {match | exclude}` *`regExp`*	Includes or excludes table rows based on the specified string found in the packet payload. **Example: set condition `content match “What are you doing Dave?”`**
property: `result {match | exclude} {success | bad-header | policy-discard | policy-refuse | socket-timeout}`	Includes or excludes table rows based on the value of the result field. See Chapter 48, “Policy objects” for information on session policies. Select one of the following, which indicates: • **success**—the packet went through. • **bad-header**—the system could not parse the header and so discarded the packet. • **policy-discard**—the system immediately discarded the packet due to a session policy firing with an action set to discard. • **policy-refuse**—the system discarded the packet, due to a session policy firing with an action set to discard, but sent a response to indicate having done so. • **socket-timeout**—the remote IP address opened a TCP connection but never sent a SIP packet, so the system closed the connection. **Example: set condition `result exclude success`**
property: `result-string {match | exclude}` *`regExp`*	Includes or excludes table rows based on any further information they may have been included in the results. **Example: set condition `result-string match “from badguy.policy”`**

url-policy

Purpose

Opens the URL policy object. It is through this object that you define the condition list to apply to entries in the URL table.

When AA-SBC receives an IM message, it scans the content for URLs. (Setting the URL determination criteria is done through the IM filtering url-list object. See Chapter 34, “IM Filtering objects” for more information.) As part of the scanning process, AA-SBC writes all URLs it finds to the URL table for use by the URL DOS engine.

Unlike the transport and SIP DOS policies, the **select** property is implicit. All action is taken based solely on the URL. When a URL meets the criteria defined, the implicit action is to drop the packet(s).

The URL policy detects when the same URL has gone through the box a specified number of times over a specified number of seconds. This is highly indicative of an IM virus that is spreading from machine to machine. When AA-SBC detects excessive appearances of a URL (for example, someone SPIMing your network with an ad for their website or a virus that self-propagates via links in IMs), it blocks future IMs containing that same URL, regardless of who the IMs appear to come from.

When opening the URL policy object, you must supply a name to identify the policy. Entering an existing name opens that policy for editing; entering a new name creates a new URL policy by that name.

Syntax

```
config vsp policies dos-policies url-policy name
```

Properties

Property name	Description
`description` *`string`*	Associates a description with the named policy. That description is then displayed when you display the configuration. Using a description allows you to attach information, such as the reason for creating the policy, without giving the policy a name that is burdensome to type each time you want to configure the rule. Enter a string up to 32 alphanumeric characters; use quotation marks if there are spaces in the description. **Example: set description “Prevent IM virus”** There is no default setting.
`admin {enabled | disabled}`	Enables or disables this URL policy. **Example: set admin enabled** The default setting is **enabled**.
`condition-list` *`conditionListReference`*	References a previously configured condition list. The list sets the rules that apply to the columns from your URL table. **Example: set condition-list vsp policies dos-policies url-condition-list 3** There is no default condition list setting.
`threshold` *`integer`*	Sets the number of unique instances (incoming packets to the system) allowed before the traffic is considered a DOS attack. This is the number of packets matching this policy (the conditions set forth in the list) that are allowed through before the system creates a dynamic rule in the kernel filter to block packets matching the identified pattern. **Example: set threshold 5** Enter a value between 2 and 1,000,000. The default setting is **10** instances.

Property name	Description
`period` *`seconds`*	Configures the interval at which you apply the URL policy to the URL table database. As a result, it also defines how many seconds of data to analyze. For example, if the period is set to 45, the system will scan the URL table every 45 seconds, and look at the last 45 seconds worth of entries. **Example: set period 45** Enter a value between 30 and 3600. The default setting is **60** seconds.
`inactivity-period` *`time`*	*Secondary property.* Defines the period of time a policy must remain inactive before its effects are removed. This is the point when packets that were being denied due to a previous policy action can again be accepted. See Setting time and time intervals for information on entry format requirements. **Example: set inactivity-timeout 600** Enter a value between 30 seconds and 30 days. The default setting is **300** seconds (5 minutes).

url-condition-list

Purpose

Opens the condition list object. It is through this list that you set the rules that apply to your URL table. The condition list defines which URL entries to examine, or alternatively, which URL entries not to examine.

Syntax

```
config vsp policies dos-policies url-condition-list name
```

Properties

Property name	Description
`description` *`string`*	Associates a description with the named condition list. Enter a string up to 32 alphanumeric characters; use quotation marks if there are spaces in the description. **Example: set description “Exclude spam IMs”** There is no default setting.
`operation {AND | OR}`	Specifies whether the conditions (properties) selected are AND’d or OR'ed together. **Example: set operation OR** The default setting is **AND**.
`url-condition {match | exclude}` *`regExp`*	Specifies which URLs to include or exclude from this policy. Match statements include those rows; exclude statements omit them. Specify a URL or regular expression that identifies a URL. **Example: set url-condition match badurl.com** There is no default setting.

20. Detect objects

Detect description

The detect settings in the AA-SBC configuration help the system to recognize the user agent (UA) sending a SIP message through the system. When a SIP UA registers with AA-SBC, the system attempts to classify the UA from data in the UA header. If AA-SBC can detect the UA type, it can add the UA data to the registration table. This data is then used by AA-SBC when establishing communication. For example, UAs may use different types of encryption. By recognizing the UA type, AA-SBC can use the correct kind of encryption when communicating with the sender (if encryption has been specified). If AA-SBC does not recognize the UA, it is possible that the registration will not complete.

Detect object summary

The following table lists and briefly describes the **detect** object. See the following chapter for other objects in the CLI hierarchy:

- Chapter 75, “Virtual system partition (VSP) objects”

Object name	Description
`detect`	Specifies user agent (UA) detection settings.

detect

Purpose

Specifies the user agent (UA) detection settings. These are the default values that AA-SBC uses to recognize SIP UAs.

Note: The default settings should only be changed if you have reason to modify the standard UA header information.

Syntax

```
config vsp detect
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables changes to the user agent detection code on the REGISTER path. Do not enable this feature unless told to do so by Technical Support. **Example: set admin enabled** The default setting is **disabled**.
windows-messenger *regExp*	Sets the regular expression to match against in the UA header. The string identifies the message as coming from a Windows Messenger UA. **Example: set windows-messenger .*RTCv1** The default setting is **.*RTC**.
office-communicator *regExp*	Sets the regular expression to match against in the UA header. The string identifies the message as coming from a Office Communicator UA. **Example: set office-communicator .* LCC/1.0** The default setting is **.*LCC/1.3**.

Property name	Description
sipura *regExp*	Sets the regular expression to match against in the UA header. The string identifies the message as coming from a Sipura UA. **Example: set sipura .*v1sipura** The default setting is .*(?i)Sipura.
snom *regExp*	Sets the regular expression to match against in the UA header. The string identifies the message as coming from a SNOM UA. **Example: set snom .*v1snom** The default setting is **.*(?i)SNOM**.
polycom *regExp*	Sets the regular expression to match against in the UA header. The string identifies the message as coming from a Polycom UA. **Example: set polycom .*v1poly** The default setting is **.*(?i)Polycom**.

Detect objects

21. Dial plan objects

Dial plan description

The dial plan allows you to direct a request to a particular gateway based on the dial prefix or domain suffix. (he dial-plan is used when AA-SBC is at the originating end of a call.) By default, dial plans apply to the following types of requests:

- INVITE
- REFER
- MESSAGE
- INFO
- OPTIONS

Note that you can change whether a message type uses the dial plan using the **sip-message-plan** property of the settings object. (You cannot change the plan type for INVITE messages as they always use a dial plan.)

When you configure components within the dial plan, AA-SBC adds those entries to the corresponding dial plan table. The system walks the tables to find matching criteria that define further selection and/or alteration. The components are considered in order of processing:

1. normalization
2. arbiter
3. route

In addition, all plans with an associated condition-list appear in the table before those without. Otherwise, AA-SBC determines the table order based on the match statement type and value. See Pattern match precedence to determine precedence of the match types.

Dial plans use a condition-list as a “first pass” filter when matching a plan entry. When AA-SBC receives an incoming request of the type listed above, it compares the request against the dial plan table entries with configured conditions, and returns a list of matching plans. The match statements within the plan components then determine the next level of filtering. If there are still multiple matches, the **priority** setting within the component determines the order of selection.

Note: Condition lists are common to several objects and are documented in Chapter 15, “Condition list objects”.

The arbiter controls the dial-plan. It functions as a “master plan,” determining which metrics to use in selecting a destination server. Note that, by default, AA-SBC uses the Request URI portion of the SIP message for dial-plan processing. You can, however, configure AA-SBC to use the To URI portion by setting the **call-routing-on** property of the server-pool object (or the carrier switch object) for the server the call came in on.

When AA-SBC receives an INVITE, it extracts the USER portion of the SIP header. If all characters are digits, AA-SBC alters the URL to “sip:xxxx@.*”. This makes the domain a wildcard match, meaning the phone number can be in any domain. However, the phone number must match the prefix specified by **request-uri-match** of the route or source-route objects. If all characters are not digits, AA-SBC does a suffix match (and the domain remains unchanged).

The configuration of the dial plans determines the entries in the call routing table. Entries are derived from configured route and source-route objects. The table determines AA-SBC lookup behavior. If a server that is referenced in a dial plan becomes unavailable, AA-SBC removes the entry from the call routing table. However, the entry remains in the configuration. Use the **show dial-plan** command to view all configured entries; use **show call-routing** to display active entries. See Call routing table for more information.

AA-SBC also maintains a normalization table and an arbitration table. The normalization table contains “scrubbed” (normalized) entries for each destination (or next-hop) server. The arbitration table maintains the arbiter configurations. AA-SBC uses this table to look up a **subscriber-match** and apply the configured rules.

Using the dial-plan configuration, AA-SBC:

1. normalizes the SIP message (using the normalization table).

2. using the normalized message, looks in the arbitration table to find the arbiter that applies to that entry.
3. executes routing lookups and routing arbitration calculations.
4. forwards the call.

See the *Net-Net OS-E – Session Services Configuration Guide* for more information.

Understanding call routing tables

AA-SBC has two internal routing tables that manage the call handling data. These tables are:

- the location routing (or call routing) table
- the dial-plan table

When an INVITE message arrives at AA-SBC, the request is forwarded based on the settings reflected in the tables. You can view the contents of the tables using the **show call-routing** and **show dial-plan** commands. REGISTER requests are handled through the registration routing table, as described in Chapter 55, “Registration plan objects”.

Call routing table

The call routing table defines how AA-SBC forwards an outgoing call. When AA-SBC receives an INVITE, it checks the call routing table for a match of the requests “to” and “from” fields. If there is a match, AA-SBC forwards the request to the matching peer and increments the reference count. The call routing table controls outgoing calls—which peer do we forward a call to?

The dial plan table contains all entries configured in the dial-plan; this includes data on dial-prefix and normalization settings. The call routing table, used for call forwarding, contains only entries with an active peer (server). If a server state becomes unavailable, the relevant route entry is removed from the call routing table. The data remains in the dial plan table, however.

The following is a sample call routing table. Entries are compiled from dial and dial-prefix plans, and server tag and domain matches.

```
NNOS-E> show call-routing
```

```
plan-name      type      match       min     pri     peer-name     fwd
---------      ----      -----       ---     ---     ---------     ---
default        phone     !*          2      100     Verizon       0
New York       phone     212!*      3      100     NNOS-E@NewYork   0
San Jose       phone     506!*       3      100     NNOS-E@SanJose   0
```

The following table describes each field and how its value is derived.

Field name	Description	Derived from...
plan name	The name of the active dial plan.	The plan created with the dial-plan or dial-prefix configuration.
type	The portion of the request to match on. If the INVITE matches the portion identified by the type, the system forwards the request to that server.	The type can be contributed from the dial-plan configuration. Types of *tag* or *domain* can be contributed from the **auto-tag-match** and **auto-domain-match** options of the server-pool-admission-control **routing-setting** property.
match	Regular expression or tag that identifies the “to directory” mapping.	The regular expression/tag can be derived from dial-plan configuration. A tag or regular expression of type *domain* can be contributed from the auto-tag-match and auto-domain-match server-pool-admission-control properties.
min	The minimum number of digits to match on in a phone prefix.	This number is specified with the **request-uri-match** (for route object) or **source-match** (for source-route object) properties when type is set to **phone-prefix**. In some cases, the system calculates a value for other types of matches based on the number of characters (including wild cards). In some cases it displays as-is. The value is only meaningful to a phone-prefix match, however.

Field name	Description	Derived from...
pri	The priority (order of preference) setting for the dial-plan entry. This property overrides the default behavior (most specific match) and sets a preference based on the **request-uri-match** (route) or **source-match** (source-route) property.	The **priority** property set with the route or source-route object.
peer-name	A statically entered peer. This is a configured server of type sip-registrar.	The dial-plan configuration.
fwd	The number of times this plan has matched an INVITE request, and the system forwarded the request.	This is a counter internal to AA-SBC.

User normalization properties

The **request-user**, **to-user**, and **from-user** normalization properties all support the same settings. See the property descriptions for an explanation of which is the effected portion of the URI for that property. These settings are described in the following table:

Setting	Description
no	No normalization applies, the URI remains unchanged.
prepend *phonePrefix*	Adds the specified phone prefix to the beginning of the current phone number.
prepend-to *resultingStringLength phonePrefix*	Adds the specified phone prefix to the beginning (portion left of the @ sign) of the URI. Specify the prefix and the resulting string length, which indicates how many total characters are in the phone number after the system prepends the phone prefix.
strip-off *numberOfCharacters*	Removes as many characters as you specify from the phone prefix. Characters are removed beginning at the far left, moving towards the @ sign.

Setting	Description
strip-off-to *resultingStringLength*	Shortens the phone prefix (portion left of the @ sign) to the number of characters you specify as the resulting string length.
replace-prefix *newPhonePrefix*	Replaces the characters identified by the match property with the characters you specify.
replace-with *newPhoneNumber*	Replaces all numbers to the left of the @ sign with the number you specify.
append *phoneExtension*	Appends the specified extension to the end of the current phone number.
thru-registration-plan	Uses the normalization settings from the VSP's registration plan (defined in the registration-plan object).

The following table provides examples of how AA-SBC prepends prefixes in various situations.

Start with...	Changes to...	Explanation
Property setting: set from-user prepend-to 10 978		
4321	9780004321	The system expands the space between the prefix and the original number with 0s.
7654321	9787654321	The system prepends the prefix, the resulting string length is correct, so no further changes are made.
555557654321	9787654321	The system prepends the prefix (3 digits), and then includes the necessary number of digits to make the resulting string length (7), starting at the @ sign and moving left. The system removes all other digits.
Property setting: set from-user prepend-to 10 3219876543210		
4321	9876543210	The system prepends as many digits as specified by the resulting string length, beginning at the @ sign and moving left. The system removes all other digits.

Using the match properties

Several of the dial plan, registration plan, and other objects use a match property to define to which calls AA-SBC should apply the configured plan. The following table identifies the objects that use the match property.

Object	Property name	Basic or extended?
Objects from the dial plan		
normalization	match	basic
source-normalization	match	basic
arbiter	subscriber-match	basic and extended
route	request-uri-match	basic
source-route	source-match	extended, but no **default** option
Objects from the registration plan		
normalization	match	basic
arbiter	subscriber-match	basic and extended
route	to-uri-match	basic
source-route	source-match	extended options only
proxy	uri-match	basic
source-route	source-match	extended options only
Objects from the carrier switch, carrier trunk-group, and server-pool server		
outbound-normalization	uri-match	basic
inbound-normalization	uri-match	basic

The following table describes each option.

Option	Apply this plan entry if the URI...	Example
Basic options		
default	...has nothing to match on. This option matches any URL that does not match a URL in a dial plan. No additional entry required. If no other option is set, use this default setting. (This property does not apply to the source-route object.)	N/A
uri-exact *uri*	... matches this URI exactly (must match the whole URI). Enter a URL (SIP URI).	sip:jdoe@abc.com or sips:jdoe@abc.com Based in the whole URI match, each will match differently.
directory [*directory*]	...indicates having arrived via this directory. Enter a reference to a configured directory.	vsp enterprise directories ldap ABC-directory
phone-exact *phoneNumber*	...matches this phone number exactly. Enter a phone number.	9785551212

Option	Apply this plan entry if the URI...	Example
phone-prefix [*phonePrefix*] [*minimalCharacters*]	...matches a field containing an alphanumeric string with the specified prefix. Enter a prefix and the minimum number of characters required in the string. For example, if you specify 10, the string in the URI must be at least 10 characters long. The prefix must display as the first characters of the URI. If you do not enter a prefix, it matches all characters. If you do not enter a number of characters, there is no minimum. Typically this field is completed with digits, but it is possible that the prefix would also contain characters. Note that if you set the regular expression in the registration-plan settings object alpha-numeric-phone-expression property and if a URI matches the regular expression, the URI will be subject to a phone prefix match.	1978
domain-exact *domainName*	...matches this domain name exactly. Enter a domain name.	abc.com
domain-suffix *domainSuffix*	...matches a field containing a domain name with the specified suffix. (The system only considers the suffix, any characters prior are ignored.) Enter a domain suffix.	anything.goes.abc.com
condition-list	...is set to condition list. The matching conditions are defined in the condition-list itself.	No arguments required
Extended options		

Option	Apply this plan entry if the URI...	Example
host *ipAddress*	...matches this source IP address.	192.168.100.100
ipnet *ipAddress/mask*	...matches this subnet.	192.168.0.0/16
server [*serverReference*]	...indicates having arrived via this referenced server.	vsp enterprise servers lcs ABC-server
carrier [*carrierReference*]	...indicates having arrived via this referenced carrier.	vsp\carriers\carrier carrier-1
gateway [*gatewayReference*]	...indicates having arrived via this referenced gateway.	vsp\carriers\carrier carrier-1\gateway gateway-1
trunk [*trunkGrpReference*]	...indicates having arrived via this referenced trunk.	vsp\carriers\carrier carrier-1\gateway gateway-1\trunk-group TG-1
local-port [*portReference*]	...indicates having arrived via this port.	50601

Pattern match precedence

The dial plan will perform a most-specific pattern match in cases where a priority assignment has not changed that behavior. Because a SIP URI can match more than one pattern—for example, sip:19998887777@company.com can match both phone prefix 1999888 as well as domain suffix company.com—you must understand the precedence level of the match patterns to correctly configure a dial plan. The following describes the precedence level AA-SBC assigns to each match pattern:

- whole URI exact match.
- directory match.
- phone prefix match (including phone exact match) applies to the URI User field and to the phone number for a TEL. See Using wildcards in the phone entry for more information.
- domain suffix match (including domain exact match) applies to the URI Host field.
- source IP subnet match (including source host match) applies to the source IP address from which AA-SBC received the SIP message.

- source peer match (including source server, source carrier, source gateway, and source trunk) applies to the source peer from which AA-SBC received the SIP message.
- default match.

Using wildcards in the phone entry

You can use a wildcard of x or X when specifying phone prefixes. Note that this is not a regular expression—this a AA-SBC-specific method of entering wildcards for use in the longest-prefix, most specific match search. A wildcard can replace any digit in the same relative position within a phone-prefix or phone-exact match. Be aware, however, that each pattern containing a wildcard can result in additional binary lookups (and further lookups for each match), which could impact performance.

The following table illustrates wildcard examples, with the text below describing the AA-SBC interpretation. In the example, there are three groups of patterns, each displaying from the least to most specific The **A** group can be categorized as pattern 0xxx, the **B** group as pattern 0000xxx, and the **C** group as null (no wildcard).

Tag	Phone entry example	Prefix or exact?	Pattern
A1 (least specific)	1xxx654	prefix	0xxx
A12	1xxx6543	prefix	
A123	1xxx6543210	exact	
B1	1987xxx	prefix	0000xxx
B12	1987xxx3	prefix	
B123	1987xxx3210	exact	
C1	1987654	prefix	Null
C12	19876543	prefix	
C123 (most specific)	19876543210	exact	

Within each group, the A/B/C 1 entry is the least specific and the A/B/C 123 entry is the most specific. AA-SBC sorts patterns alphanumerically, resulting (in the example) in a search order of pattern group Null, 0000xxx, and finally 0xxx.This order establishes the lookup precedence order. The following table shows the results of some sample lookups, with varying lookup patterns available. (A change in the previous number is indicated by bold italic.)

Entry...	Matches...	Phone entry example
19876543210	C123	1987654
1987654321*1*	C12	19876543
1987654*1*210	C1	19876543210
Without the null pattern...		
19876543210	B123	1987xxx
1987654321*1*	B12	1987xxx3
1987654*1*210	B1	1987xxx3210
Without the null or the 0000xxx pattern...		
19876543210	A123	1xxx654
1987654321*1*	A12	1xxx6543
1987654*1*210	A1	1xxx6543210

Dial plan object summary

The following table lists and briefly describes the **dial-plan** object. See the following chapters for other objects in the CLI hierarchy:

- Chapter 15, “Condition list objects”
- Chapter 62, “Session configuration objects”
- Chapter 75, “Virtual system partition (VSP) objects”

Object name	Description
dial-plan	Opens the dial-plan configuration object.
dial-prefix	Applies a custom session configuration based on a dial prefix found in either the To or Request URI of the SIP header.
normalization	Facilitates routing lookup by normalizing the SIP message.
condition-list	Sets matching conditions for application of the normalization plan. See Chapter 15, “Condition list objects”.

Object name	Description
source-normalization	Facilitates routing lookup by normalizing SIP messages, with matches based on the source IP address.
arbiter	Configures an ordered set of rules to influence the routing arbitration decision.
condition-list	Sets matching conditions for application of the arbiter plan. See Chapter 15, “Condition list objects”.
route	Configures match criteria for route selection based on the Request URI.
condition-list	Sets matching conditions for application of the route plan. See Chapter 15, “Condition list objects”.
source-route	Configures match criteria for route selection based on the source IP header or From URI rather than the Request URI.
condition-list	Sets matching conditions for application of the source-route plan. See Chapter 15, “Condition list objects”.

dial-plan

Purpose

Opens the dial plan object through which you define the phone numbers and suffix entries and, optionally, the exceptions for outgoing phone calls.

Syntax

```
config vsp dial-plan
```

Properties

None

dial-prefix

Purpose

Applies a custom session configuration based on a dial prefix found in either the Request URI of the INVITE or the To header of a REGISTER in the SIP header. For example, if you set a recognizable sequence for originating a phone call, your session configuration could initiate call recording when AA-SBC recognized that dial prefix in the SIP header.

Once you have created an entry, you can set session configuration characteristics to apply to calls matching the prefix by referencing a previously configured session-config-pool object. For details of the session configuration objects, see Chapter 18, “Default session configuration objects” descriptions.

Syntax

```
config vsp dial-plan dial-prefix entryName
```

Properties

Property name	Description
description *string*	Associates a text string with a dial-prefix configuration. The string displays in some event logs and status providers to help identify the target. **Example:** set description E911server There is no default setting.
admin {enabled | disabled}	Enables or disables this dial-prefix plan and its associated session configuration. **Example: set admin enabled** The default setting is **enabled**.

Property name	**Description**
routing-tag {wildcard | named *string* | anonymous}	Labels this dial-prefix profile for matching against routing tags. For example, if the system picks up an ingress routing tag, it only selects dial plans that have a matching tag configured. • **wildcard**—there is no label on the route (matches all tags). • **named**—labels the route with the specified string. • **anonymous**—labels the route with the tag “anonymous” (the literal string). **Example: set routing-tag named rate3** The default setting is **wildcard**.
dial-prefix *regExp*	Specifies the portion of the dial prefix to match and act on to initiate a custom session configuration. If an incoming call matches the prefix specified by the entry, the system applies the entry session configuration. This custom session configuration is configured as a subobject of the entry or by referencing a session-config-pool using the **session-config-pool-entry** property. Enter a dial prefix to match on when creating the entry. The prefix must display as the first characters of the URI. **Example: set dial-prefix *11** There is no default setting.
session-config *sessionConfigReference*	Specifies a previously configured entry in the session-config-pool object. If this property is set, the system applies the session configuration characteristics to all calls matching this dial prefix plan entry. Alternatively, you can set session configuration characteristics as a subobject of the **dial-prefix** object. **Example: set session-config-pool-entry vsp session-config-pool entry record** There is no session configuration applied by default.

normalization

Purpose

Initiates normalization for matching SIP messages. (Normalization applies to INVITE, MESSAGE, INFO, and OPTIONS requests.) This normalization occurs before arbitration or routing lookup (facilitating the lookup).

See User normalization properties for information on the changes that normalization can effect.

Note: In most cases it is preferable to use the outbound-normalization and inbound-normalization objects that are available under the switch, trunk-group, or server objects. The dial-plan normalization cannot be used when call forking or failover is in use, as it only provides normalization on the primary path. It is best used for normalizing endpoint information.

Understanding phone synchronization

When a phone registers with AA-SBC, it registers using its full telephone number and domain name. However, some PBXs abbreviate the numbers in the URI, causing lookup problems in the location cache. For example, when the caller and callee are in the same BroadWorks group, the server may just use the extension.

AA-SBC first applies normalization, which can include phone synchronization, to the URI when processing a call. After normalizing the message, the system performs a location cache lookup on the Request, From, and To URI of the INVITE.

Phone synchronization compares the From header to the Request/To header. The longer header, if the total number of digits is the same as in phone synchronization configuration, is considered the complete number; the shorter header is the abbreviated version. When you set phone synchronization, you specify the total digits for a complete phone number. The system expands the number it received in the shorter header so that it meets that specified length, prepending digits from the header that contained the complete URI.

Phone synchronization does not apply to global phone numbers (a plus sign present in the Request URI) or nonnumeric users (e.g., LCS and Sametime).

Syntax

```
config vsp dial-plan normalization string
config vsp dial-plan source-normalization string
```

Properties

Property name	Description
description *string*	Associates a text string with a normalization plan. The string displays in some event logs and status providers to help identify the target. **Example: set description E911server** There is no default setting.
admin {enabled | disabled}	Enables or disables this normalization plan. When **enabled**, the system provides normalization for matching SIP messages. When **disabled**, you can configure the plan properties but the system does not apply it. **Example: set admin enabled** The default setting is **enabled**.
routing-tag {wildcard | named *string* | anonymous}	Labels this normalization profile for matching against routing tags. For example, if the system picks up an ingress routing tag, it only selects dial plans that have a matching tag configured. • **wildcard**—there is no label on the route (matches all tags). • **named**—labels the route with the specified string. • **anonymous**—labels the route with the tag “anonymous” (the literal string). **Example: set routing-tag named rate3** The default setting is **wildcard**.

Property name	Description
condition-list-match-secondary {false | true}	Specifies whether a condition list match should also be required for the specified match property. If the match property is set to something other than condition-list, you can set this property to **true** to use a condition list in addition to the type selected. In that case, the call must match both the primary key and the condition list. **Example: set condition-list-match-secondary true** The default setting is **false**.
priority *value*	Specifies an order of preference for this normalization plan. Often, a number or URI will match multiple normalization entries. By default, the system uses the most specific match. Use this property to override that default behavior and set a preference based on the **match** property. See Assigning priority for more information. **Example: set priority 200** The default setting is **100**.
apply-to-headers {request-uri | to-header | from-header}	Specifies the header type to which the system applies the **match** property. Headers containing the configured match in the selected method(s) are then normalized according to this plan. You can enter multiple header types; separate them by a plus sign (+) with no spaces. **Example: set apply-to-headers request-uri+from-header+ to-header** The default setting is **request-uri**.
alter-tel-scheme {no | yes *domainName*}	Specifies whether to modify the Tel column of the SIP header. If set to **no**, no changes occur. If set to **yes**, the system appends the specified domain name to the telephone number and changes the Tel scheme to the SIP scheme. **Example: set alter-tel-scheme yes abc.com** The default setting is **no**.

Property name	Description
enum-operation {enabled | disabled | asynchronous}	Specifies whether the contact with the ENUM server should be synchronous or asynchronous. Select either: • **enabled**—the system performs a synchronous lookup. • **disabled**—the system does not include ENUM lookup as part of the normalization plan. • **asynchronous**—the system performs an asynchronous lookup. **Example: set enum-operation enabled** The default setting is **disabled**.
enum-apply-request-result-to-contact {enabled | disabled}	*Secondary property.* Specifies whether to normalize the Contact URI. When **disabled**, the Contact URI is not modified. When **enabled**, the system normalizes the Contact URI. It does so by applying, to the Contact URI, the regular expression match string and replacement results returned in the NAPTR record for the Request URI. When enabling this property, you must also set the **enum-operation** property to **enabled**. In addition, because by default the system changes the Contact URI to itself on the outbound leg, you must configure the session-config contact-uri-settings-out-leg to preserve the normalization changes. **Example: set enum-apply-request-result-to-contact enabled** The default setting is **disabled**.

Property name	**Description**
enum-server *name* [*domainName*] [any | UDP | TCP | TLS] [*port*] [*preference*]	Specifies the name(s) of the external ENUM server used for normalization lookups. The system selects a server to query based on this configuration, resulting in phone number-to-user name conversions. When identifying the server enter: • a host name or IP address. There is no default. • the domain name of the ENUM server, appended to the SIP phone number string. The default is e164.arpa • protocol to use to contact the ENUM server. The default protocol is UDP. • the port over which the system sends ENUM requests. The default is 53. • the preference for this server. The default is 100. **Example: set enum-server 192.168.10.10 rfc2916.net 5001 UDP 110** There is no default setting for the name. The default domain name is **e164.arpa**, default protocol is **UDP**, default port number **53**, and the default preference is **100**.
synchronize-phone-group {no | yes [*totalDigits*]}	Enables phone synchronization based on a dial-plan match. When you set phone synchronization to **yes**, you also specify the total digits for a complete phone number. Synchronization is disabled by default. See Understanding phone synchronization for complete information. **Example: set synchronize-phone-group yes 10** The default setting is **no**. If set to **yes**, the default number of digits is 10.

Property name	Description
apply-to-methods *messageType*	Specifies the message type to which the system applies the **match** property. Those messages containing the configured match in the selected header(s) are then normalized according to this plan. When you modify this value, the system overwrites the current setting with only the message types you specify. For example, if set to the default (all selected) and you enter **OPTIONS**, the system will match against only the OPTIONS portion of the header. Enter multiple message types separated by a plus sign (+) with no spaces. Note that this property is overridden by the values configured with the **sip-message-plan** property of the settings object. If a message type is assigned to registration-plan in that property, you cannot control normalization here. **Example: set apply-to-methods INVITE+REFER** The default setting is **INVITE+REFER+MESSAGE+ INFO+OPTIONS**.
request-user *setting*	Sets the type of normalization that the system applies to outgoing calls to a provider (to the USER field of the Request URI). See User normalization properties for property setting options and descriptions. **Example: set request-user prepend 1978** The default type setting is **no** (no normalization applied).
to-user *setting*	Sets the type of normalization that the system applies to outgoing calls to a provider (to the USER field of the To URI). See User normalization properties for property setting options and descriptions. **Example: set to-user strip-off-to 10** The default type setting is **no** (no normalization applied).

Property name	Description
from-user *setting*	Sets the type of normalization that the system applies to outgoing calls to a provider (to the USER field of the From URI). See User normalization properties for property setting options and descriptions. **Example: set from-user thru-registration-plan** The default type setting is **no** (no normalization applied).
normalize-again {enabled | disabled}	Enables cascading normalization on the SIP message. If **enabled**, when the system completes normalizing the message, it then compares the result to the plans in the normalization table and if it finds a match, applies normalization again. This process continues until the message matches a plan that is disabled. **Example: set normalize-again enabled** The default setting is **disabled**.
match *type* [*string*]	Specifies which SIP messages have normalization applied. Entries matching the type defined are then normalized. **Example: set match phone-prefix 978823** There is no default setting.

source-normalization

Purpose

Initiates normalization for matching SIP messages based on matching the source IP address. (Normalization applies to INVITE, MESSAGE, INFO, and OPTIONS requests.) This normalization occurs before arbitration or routing lookup (facilitating the lookup).

Syntax

```
config vsp dial-plan source-normalization string
```

Properties

For a complete description of the **source-normalization** properties, see the normalization object.

arbiter

Purpose

Configures an ordered set of rules to influence the routing arbitration decision for those calls meeting dial plan match criteria. These rules configure different metrics, which AA-SBC uses to select where to forward inbound SIP calls. When AA-SBC receives a SIP call, it makes a determination where to forward the call (to the next hop) based on a routing arbitration decision. This is necessary for a given destination SIP server because multiple carriers may be available to route the call.

The arbiter function is usually subscriber based. This means that arbitration is applied based on who originated the call (the source), for example, indicated in the URI in the From header. In addition, when a subscriber match is found, the arbiter can allow matching of the request-URI or to-URI (defined in the server configuration, **call-matching-on** property), allowing, for example, business calls to maximize performance while personal calls are routed economically. Note that by using a condition-list you can match on either source or destination of a call.

AA-SBC uses the longest prefix/suffix match for lookups within the **arbiter** rules. See Finding the most-specific entry for more information.

If the arbiter configuration results in a route selection that does not meet the rule criteria, AA-SBC responds to the downstream server with a “486 Busy” message, indicating that the route was resolved but that the server was unavailable. For example, a subscriber attempts a call while having a least-cost rule and the highest allowable rate set to 10 cents per second. All PSTN gateways supporting that rate are overloaded or down. Only a premium PSTN gateway, at a cost of 50 cents per second, is available. AA-SBC responds with 486 Busy and declines the call.

If AA-SBC does a lookup in the arbitration table and finds no entries, it uses “factory” default settings. These are:

- Use the **best-match** setting for **arbiter-apply**

- Use the most-preferred, least-calls, and least-load routing calculation algorithms.

Routing arbitration rules

The routing arbitration **rule** property sets the criteria by which AA-SBC selects the server to which it forwards calls. The system updates the statistics with each call routed to a server. You can set as many rules as you wish for each **arbiter** object. Keep in mind that the system evaluates the rules in the order they are created. To re-order the rules, use the move command.

When all values are the same for a rule, the arbiter skips to the next configured rule. For example, consider a configuration with the first rule for least-cost and the second for weighted call average. If all costs are the same, the arbiter skips to the next configured rule (weighted-call-average). In general, the arbiter starts at the first configured rule and moves down the rule list until if finds a value difference. If it reaches the end of the list before finding a difference, AA-SBC uses the server configuration order to decide which server to use.

The following table describes the routing algorithms that are available for routing arbitration.

Rule option	AA-SBC uses the server...
most-preferred	...you selected by configuring the server preference. That value is set with the server-pool server-pool-admission-control object **preference** property. If there are multiple carriers marked most-preferred with the same preference, the system uses the next rule in the arbiter to make a forwarding determination.
least-cost	...with the lowest routing cost metric to that destination. This metric is set via the carrier (and carrier switch, gateway trunk-group) **flat-rate** property or the gateway (and gateway trunk-group) rate-plan object.
best-mos	...link having the best media mean opinion score (MOS) value. This metric measures the audio quality for the listener on the phone. See the *Net-Net OS-E – Session Services Configuration Guide* for information on MOS calculations.
best-asr	...link having the highest answer-to-seizure ratio (ASR). This metric measures the number of calls made versus the number of calls answered.
least-pdd	...link having the lowest post-dial delay (PDD). This metric measures the amount of time elapsed, measured from after dialing a number to before receiving a ringing/busy answer.

Rule option	AA-SBC uses the server...
best-acd	...link having the best average call duration (ACD). This metric measure the length of an average call.
least-load	...that is most available based on having the most bandwidth available. You must set the master services server-load object for the system to calculate server loads.
least-calls	...that is most available based on the managing the least number of calls at that time.
weighted-call-average	...that is selected based on a weighted round robin algorithm. Using the **max-number-of-concurrent-calls** property set in the server-pool-admission-control object, the system calculates which server should receive the call. It derives a next-percentage-of-calls value (1% of the maximum concurrent calls), and establishes a call percentage value from there. The system then selects the next server by comparing the call percentage value across servers. See Weighted call average example, below.
weighted-round-robin	...that is dynamically selected based on the call activity across all servers in the server pool. While the weighted-round-robin algorithm is similar to the weighted-call-average based on the active call percentage of the **max-number-of-concurrent-calls** property in the server-pool-admission-control object, weighted round robin is less deterministic in selecting the server. In cases where all servers are processing the same relative percentage of their maximum calls, the weighted-round-robin algorithm will select server at random based on availability rather than current utilization percentage. Using the weighted-round-robin results in a more equal distribution of calls among servers in the pool.

Weighted call average example

Weighted call average is calculated based on the number of connected calls to a server and the server **max-number-of-concurrent-calls** setting. The algorithm divides the **max-number-of-concurrent-calls** by 100 to determine how many calls must connect before sending calls to the next server in the pool.

For example, if server A has **max-concurrent-calls** set to 1000, and server B has it set to 500, then the server A “bucket” size is 10 and server B is five. This means that server A must have 11 calls (bucket size plus one) connected before AA-SBC routes calls to server B. Server B must have six calls connected before AA-SBC routes back to server A again. In other words, AA-SBC implements a round-robin distribution through the server pool based on the configuration order, but each server must fill its bucket-plus-one before the system moves on to the next server. Note that if the **max-number-of-concurrent-calls** setting is less than 100, then the bucket size is set to 0 (each server gets 1 call).

Syntax

```
config vsp dial-plan arbiter string
```

Properties

Property name	Description
description *string*	Associates a text string with a dial-plan arbiter. The string displays in some event logs and status providers to help identify the target. **Example: set description E911server** There is no default setting.
arbiter-apply {best-match | joined-matches}	Specifies whether to apply the arbiter rules to the best match or to all matches in the routing table lookup. Select either: • **best-match**—by default, the arbiter rules apply to the most specific match. Or, if configured, to the route with the lowest priority (set with the **priority** property of the route object). In the event of a tie, the system selects the most specific. • **joined-matches**—the system merges all routes that match the dial plan and then sorts them according to the routing arbitration rule specified in the **rule** property. This value is then used by the **handle-response** property of the server-pool object, when set to **try-next-route**. **Example: set arbiter-apply joined-matches** The default setting is **best-match**.

Property name	Description
max-call-hunting-options *integer*	Specifies the maximum number of gateways and/or trunk-groups over which the system can hunt for a call should a gateway/trunk failure occur. **Example: set max-call-hunting-options 50** Enter any number greater than 1; the default setting is **100**.
call-hunting-type {none | sequential | parallel}	Determines the order or method in which the system forwards the call to the next-hop gateway. Unless set to **none**, this setting takes precedence over any forking settings set by the server object **call-hunting-type** property. • **none**—the system forwards the call to the latest binding for the Request URI. • **sequential**— if there are two or more servers in a server pool, the system first tries the primary and then the secondary. • **parallel**—when the system receives a call, it creates two call legs and forwards to both the primary and secondary server. When one server responds, the system disconnects the call with the other server. **Example: set call-hunting-type sequential** The default setting is **none**.

Property name	Description
call-routing-on {request-uri | to-uri | as-is}	Specifies whether the system does routing or location lookups based on the Request URI, the To URI, or an alternate setting. By default, the system performs lookups on the Request URI. Change this setting, for example, when routing information is not available in the Request URI but it is available in the To URI. This setting can also be configured in the server-pool object. If values are set in both this and the server-pool, the arbiter settings take precedence. • **request-URI**—the Request URI, which contains the hop-by-hop destination for the call. • **to-uri**— the To URI, which contains the final destination of the call. • **as-is**—the Request URI (the default) or the value set for this property in the server-pool object. **Example: set call-routing-on request-uri** The default setting is **as-is**.
min-calls-apply-constraints *integer*	Specifies a minimum number of calls that must be active before quality constraints are applied. The system does not route based on quality metrics until the severs and routes have reached the minimum set with this property. The system then has enough time and data to calculate meaningful values before routing. The constraints only apply to QoS routing arbiter rules (mos, acd, asr, and pdd). **Example: set min-calls-apply-constraints 150** Enter any number between 1 and 65535; the default priority setting is **100**.
max-cost {unlimited | *centsPerSecond*}	Sets the maximum rate, in cents-per-second, that a call can cost. If this property is set to unlimited, there is no limit to the calling rate. **Example: set max-cost 9** The default setting is **unlimited**.

Property name	Description
min-available-bandwidth *kbps*	Sets a maximum threshold of available bandwidth for a server, limiting the amount of traffic forwarded to that downstream server. For each connected call to a server, the system calculates the bandwidth used based on the CODEC. If all calls to server exceed this bandwidth limit, the system ceases routing to that server until the bandwidth again becomes available. A value of 0 disables the functionality. **Example: set min-available-bandwidth 1000** The default setting is **0** kilobits-per-second.
call-routing-lookup {calling-group | dial-plan | dial-plan-tagged}	Specifies which table(s) the system should use for route lookup when routing a call. You can enter multiple routing lookup options. The system searches the route in all tables specified, and then selects a route based on the criteria specified in the arbiter configuration. If you select **dial-plan**, the system performs a call routing table lookup. If you select **calling-group**, the system performs a lookup in the referenced calling-group table. **Example: set call-routing-lookup calling-groups “vsp calling-groups group 1"** The default setting is **dial-plan**.
session-config *sessionConfigReference*	Specifies a previously configured entry in the session-config-pool object. If this property is set, the system applies the session configuration characteristics to all calls matching this arbiter entry. **Example: set session-config-pool-entry vsp session-config-pool entry rule1** There is no session configuration applied by default.

Property name	Description
apply-to-methods *messageType*	Specifies the message type to which the system applies the **subscriber-match** property. Those messages containing the configured match in the selected header(s) are then subject to routing arbitration according to this plan. When you modify this value, the system overwrites the current setting with only the message types you specify. For example, if set to the default (all selected) and you enter **REFER**, the system will match against only the REFER portion of the header. Enter multiple message types separated by a plus sign (+) with no spaces. Note that this property is overridden by the values configured with the **sip-message-plan** property of the settings object. If a message type is assigned to a registration-plan in that property, you cannot control routing arbitration here. **Example: set apply-to-methods INVITE+REFER** The default setting is **INVITE**.
rule *algorithm*	Enters rules into the arbiter configuration. Enter as many rules as you wish. If you do not set any rules, the system uses the default settings. **Example:** set rule least-cost 15 There is no default setting.
routing-tag {wildcard | named *string* | anonymous}	Labels this arbiter profile for matching against routing tags. For example, if the system picks up an ingress routing tag, it only selects dial plans that have a matching tag configured. • **wildcard**—there is no label on the route (matches all tags). • **named**—labels the route with the specified string. • **anonymous**—labels the route with the tag “anonymous” (the literal string). **Example: set routing-tag named rate3** The default setting is **wildcard**.

Property name	Description
subscriber-match *type* [*string*]	Specifies what to match in the USER and/or HOST fields of the FROM URI in order for the system to apply the plan configuration to calls containing the prefix. The **subscriber-match** value defines the criteria for matching entries in the arbitration table; the applicable arbiter is then applied to matches, determining the calculation the system performs. **Example: set subscriber-match server "vsp enterprise server lcs lcs-server"** There is no default setting.
condition-list-match-secondary {false | true}	Specifies whether a condition list match should also be required for the specified **subscriber-match** property. If the match property is set to something other than condition-list, you can set this property to **true** to use a condition list in addition to the type selected. In that case, the call must match both the primary key and the condition list. **Example: set condition-list-match-secondary true** The default setting is **false**.
admin {enabled | disabled}	Enables or disables this arbiter entry and its associated session configuration. **Example: set admin disabled** The default setting is **enabled**.
priority *value*	Specifies an order of preference for this dial-plan arbiter entry. Often, a number or URI will match multiple arbiter entries. By default, the system uses the most specific match. Use this property to override that default behavior and set a preference based on the **subscriber-match** property. See Assigning priority for more information. **Example: set priority 50** Enter any number greater than 1; the default priority setting is **100**.

route

Purpose

Configures AA-SBC to make call routing/forwarding decisions based on information in the Request URI. Use the source-route object to make routing decisions based on the IP packet header or the From URI of the SIP message. With the source-route object, a route is selected based on the source while this object selects based on the destination. AA-SBC checks for a source-route match first, then a route match.

The route configuration specifies the portion of the Request URI (dial prefix, domain suffix, condition list criteria) to match on to initiate direction of the call to a particular gateway. If an outgoing call matches the **request-uri-match** value specified in the entry, AA-SBC applies the entry session configuration to the call.

Dial plans apply session configuration in the following way:

1. If there is a session configuration added specifically for the dial plan, AA-SBC uses those settings. This is a custom session configuration that you configure as a subobject of the dial-plan entry.
2. If there is a reference to session configuration that is part of a pool, then AA-SBC uses the settings in the referenced object. These settings are configured in the VSP session-config-pool object.
3. If both cases are true, AA-SBC merges the two session configurations. If there are settings that overlap (and contradict each other), AA-SBC uses the settings from the referenced **session-config-pool-entry**.
4. Otherwise, no session configuration applies to the dial-plan.

For details of the session configuration objects, see the Chapter 62, “Session configuration objects” descriptions.

Finding the most-specific entry

AA-SBC uses a longest-prefix match lookup to match the most specific entry. If a gateway becomes unavailable, AA-SBC finds the next longest match and forwards the call to that gateway. The following table describes the order of precedence for the **request-uri-match** (from the route object) or **source-match** (from the source-route object) values.

Most to least specific	From	Example
1. `condition-list`	Configuration	Multileveled configuration of a condition list is the most specific match.
2. `uri-exact`	URI (To or From header)	Always most specific if a condition list is not configured (e.g., sip:joe@abc.com)
3. `directory`	URI (To or From header)	vsp\enterprise\directories\notes-directory East is more specific than a phone number.
4. `phone-exact`	User (To or From header)	19788235233 is more specific than phone-prefix 1978
5. `phone-prefix`	User (To or From header)	1978 is more specific than abc.com
6. `domain-exact`	Host	abc.com is more specific than domain-suffix users.abc.com
7. `domain-suffix`	Host	users.abc.com is more specific than host 192.168.100.100
8. `default`	N/A	If no other match applies, this is the least specific route match (route only).
The following options apply to source-route only.		
9. `host`	Subnet	192.168.100.100 is more specific than ipnet 192.168.0.0/16
10. `ipnet`	Subnet	192.168.0.0/16 is more specific than port 5060
11. `local-port`	Peer	Port 5060 is more specific than a server reference
12. `server`	Peer	vsp\enterprise\server\lcs lcs-1 is more specific than a carrier reference
13. `carrier`	Peer	vsp\carrier\carrier-1 is more specific than vsp\carrier carrier-1\gateway GW-1

Most to least specific	From	Example
14. `gateway`	Peer	vsp\carrier\carrier-1\gateway GW-1 is more specific than vsp\carrier\carrier-1\gateway GW-1\trunk-group TG-1
15. `trunk`	Peer	Trunk-group is least specific. If none of these apply, the system uses the default

Assigning priority

An incoming call may match multiple dial plans. To control which plan is used you can specify a preference with the **priority** property. By default, all dial plans have a priority of 100, which means the longest prefix match is effective. In some cases, however, you may want to make a less-specific entry more preferred. To do this, you would assign a higher priority (lower value) to that entry, or, you could assign a lower priority (higher value) to the more specific entry. For example:

For example, *sip:9785551212@company.com* matches the following dial-plans:

Prefix	Priority
978.*	100
978555.*	100
.*@company.com	100

By default, the match occurs on 978555.*, because it is the most specific, and AA-SBC uses that plan. However, by changing priorities:

Prefix	Priority
978.*	50
978555.*	100
.*@company.com	100

You can change the plan used. In this case, AA-SBC uses 978.* instead. It would also use 978.* if priorities were configured as:

Dial plan objects

Prefix	Priority
978.*	100
978555.*	300
.*@company.com	100

You could force AA-SBC to use the least specific entry, **@company.com*, like this:

Prefix	Priority
978.*	100
978555.*	100
.*@company.com	10

Syntax

```
config vsp dial-plan route string
config vsp dial-plan source-route name
```

Properties

Property name	Description
description *string*	Associates a text string with a dial plan. The string displays in some event logs and status providers to help identify the target. **Example: set description E911server** There is no default setting.
admin {enabled | disabled}	Enables or disables this dial route entry plan and its associated session configuration. **Example: set admin enabled** The default setting is **enabled**.
condition-list-match-secondary {false | true}	Specifies whether a condition list match should also be required for the specified match property. If the match property is set to something other than condition-list, you can set this property to **true** to use a condition list in addition to the type selected. In that case, the call must match both the primary key and the condition list. **Example: set condition-list-match-secondary true** The default setting is **false**.
priority *value*	Specifies an order of preference for this dial-plan entry. Often, a number or URI will match multiple dial-plan entries. By default, the system uses the most specific match. Use this property to override that default behavior and set a preference based on the **request-uri-match** property. See Assigning priority for more information. **Example: set priority 50** Enter any number greater than 1; the default priority setting is **100**.

Property name	Description
location-match-preferred {up-to-outbound-peer | best-effort | exclusive | except-from-server | no}	Specifies how the system should forward a call if it finds a location cache match for the endpoint. Select either: • **up-to-outbound-peer**—if the next-hop peer is a **provider** (set with the server-pool-admission-control **service-type** property), the system forwards the call to that provider peer (provided that the call is not originated from that peer). If it is of **service-type** external or internal, and there is a location cache match for the endpoint, the system forwards the call directly to the endpoint. If there is no match, it forwards the call to the next-hop peer. • **best-effort**—the system always attempts, regardless of the **service-type** setting, to forward the call to the endpoint if there is a location cache match, or to the next-hop peer. If calling to the endpoint fails or times out, the call may, if enabled in the **call-hunting-type** property of the arbiter, be sequentially forked to the next-hop peer. • **exclusive**—the system always attempts, regardless of the **service-type** setting, to forward the call to the endpoint if there is a location cache match, or to the next-hop peer. If calling to the endpoint fails or times out, the call is *not* sequentially forked to the next-hop peer, even if sequential forking is enabled. • **except-from-server**—if the call was received from a server, the system tries the location cache first for an endpoint match. Otherwise, it uses the dial-plan to forward the call. • **no**—the system never forwards the call directly to the endpoint (even if it has a location match for it), regardless of the **service-type** setting. **Example: set location-match-preferred best-effort** The default setting is **up-to-outbound-peer**.

Property name	Description
action {forward | block | redirect}	Specifies how the system processes any INVITE it receives that matches the dial-plan entry. Select either: • **forward**—the system forwards the INVITE to the server specified in the header. • **block**—the system rejects calls matching this dial plan (responds with a SIP message of status busy). • **redirect**—the system sends a response to the client with instructions to send the invite to a different server. The server noted in the response is the one that you configure with the **peer** property of this object. **Example: set action redirect** The default type setting is **forward**.
peer {none | server | carrier | exchange| switch | trunk | hunt-group | calling-group} *reference*	Specifies to which server the system should forward the call. Enter the reference path to a previously configured server or group of the type specified. **Example:** `set peer server “vsp enterprise servers sip-gateway companyABC”` `set peer trunk “vsp carriers carrier bws gateway gw100 trunk-group china”` The default setting is **none**.

Property name	Description
apply-to-methods *messageType*	Specifies the message type to which the system applies the **request-uri-match** (route object) or **source-match** (source-route object) property. Those messages containing the configured match in the selected header(s) are then forwarded according to this plan. When you modify this value, the system overwrites the current setting with only the message types you specify. For example, if set to the default (all selected) and you enter **OPTIONS**, the system will match against only the OPTIONS portion of the header. Enter multiple message types separated by a plus sign (+) with no spaces. Note that this property is overridden by the values configured with the **sip-message-plan** property of the settings object. If a message type is assigned to registration-plan in that property, you cannot control forwarding here. **Example: set apply-to-methods INVITE+MESSAGE** The default setting is **INVITE+REFER+MESSAGE+ INFO+OPTIONS**.
request-user *setting*	Sets the type of normalization that the system applies to outgoing calls to a provider (to the USER field of the Request URI). See User normalization properties for property setting options and descriptions. **Example: set request-user prepend 1978** The default type setting is **no** (no normalization applied).
to-user *setting*	Sets the type of normalization that the system applies to outgoing calls to a provider (to the USER field of the To URI). See User normalization properties for property setting options and descriptions. **Example: set to-user strip-off-to 10** The default type setting is **no** (no normalization applied).

Property name	Description
from-user *setting*	Sets the type of normalization that the system applies to outgoing calls to a provider (to the USER field of the From URI). See User normalization properties for property setting options and descriptions. **Example: set from-user thru-registration-plan** The default type setting is **no** (no normalization applied).
admission-control {enabled | disabled}	Specifies whether the system considers downstream server capacity when forwarding a call using this route. The system tracks the number of concurrent calls for each server. If this property is **enabled**, the system does not forward calls using the route if the server limit has been reached and instead sends a “503 Service Unavailable” message. (Note that if an arbiter is configured, admission control also considers quality metrics when routing.) If **disabled**, the system does forward calls from the route. (Set the call limit with the **max-number-of-concurrent-calls** property.) **Example: set admission-control enabled** The default setting is **disabled**.
max-number-of-concurrent-calls *integer*	Specifies the number of calls allowed to use this route at one time. When this value is reached, the system will not forward calls on this route until the value drops below the threshold. **Example: set max-number-of-concurrent-calls 1500** Enter a value between 0 and 1,000,000; the default is **100** calls. A value of 0 causes the system to decline all calls and registrations.
session-config *sessionConfigReference*	Specifies a previously configured entry in the session-config-pool object. If this property is set, the system applies the session configuration characteristics to all calls matching this dial plan entry. **Example: set session-config-pool-entry vsp session-config-pool entry 1** There is no session configuration applied by default.

Property name	Description
route object only request-uri-match *type* [*string*]	Specifies what to match in the USER and/or HOST fields of the REQUEST URI in order for the system to apply the entry session configuration to calls containing the prefix. **Example: set request-uri-match phone-prefix 978 10** The default type setting is **phone-prefix**. There is no default for the prefix itself. The default minimum digits is 0 (or **as-is**), meaning that the system uses the actual length of the string.
source-route object only source-match *type* [*string*]	Specifies the source of the SIP message, matching on the From URI. For all traffic from this source, the system sets the next-hop server (defined with the **peer** property) for those that match this configured source. Note that the **default** option is not available for **source-match**, as the route and source-route objects use the same lookup table, and can therefore only have a single default setting. **Example: set source-match ipnet 192.168.0.0/16** There is no default setting.

Property name	Description
max-bandwidth {unlimited | *kbps*}	*Secondary property.* Specifies the amount of bandwidth the system allocates to this route. For a SIP server, the default value is **unlimited** or the server uplink bandwidth. For example, if the uplink is GigE, then bandwidth is 1 million kbps. When the system reaches the maximum bandwidth limit for a server, it rejects calls until bandwidth use drops below the maximum. Note that the bandwidth usage value is based not on the actual traffic on the wire, but on a calculation done by the system. The calculation uses the value associated with the first known CODEC identified in the SDP for a usage rate. If there is not a known CODEC, or the value has not yet been determined from the SDP, the system uses the **default-session-bandwidth** value from the session configuration media object. Set a specific bandwidth if you are using, for example, a TDM trunk or PSTN gateway with limited bandwidth. For a PSTN trunk, the usual capacity is DS0 (64 kbps bandwidth). If a gateway has 8 trunks, then the gateway has 512 kbps bandwidth. **Example: set max-bandwidth 512** The default setting is **unlimited**.
emergency {true | false}	*Secondary property.* Specifies whether a call matching the dial prefix should be handled without limitation. If set to **true**, matching calls will not be subject to emission and admission controls. **Example: set emergency true** The default setting is **false**.

Property name	Description
response-code *code*	*Secondary property.* Sets the response code that the system sends to an endpoint when the **action** property is set to **accept** or **block**. (2xx response codes indicate success; change this value if the action is **block** and you have configured a **response-string**.) **Example: set response-code 201** The default response code is **200**.
response-string *string*	*Secondary property.* Sets the response string that the system sends to an endpoint when the **action** property is set to **accept** or **block**. **Example: set response-string "REGISTER was blocked"** There is no default response string.

source-route

Purpose

Configures AA-SBC to make call routing/forwarding decisions based on information in the IP packet header or the From URI of the SIP message. (Use the route object to make routing decisions based on Request URI information.) With the route object, a route is selected based on the destination while this object selects based on the source. AA-SBC checks for a source-route match first, then a route match.

The source-route configuration specifies the portion of the IP header or From URI to match on to initiate direction of the call to a particular gateway (set with the **peer** property). If an outgoing call matches the **source-match** value specified in the entry, AA-SBC applies the session configuration for that entry to the call.

For detailed information on how AA-SBC selects routes, see Finding the most-specific entry and Assigning priority in the route description. For details of the session configuration objects, see Chapter 62, "Session configuration objects".

Syntax

```
config vsp dial-plan source-route name
```

Properties

See the route object for property descriptions.

22. Diameter client and server objects

Diameter description

The Diameter protocol, as described in *RFC 3588, The Diameter Base Protocol*, provides authentication, authorization and accounting (AAA) services for applications such as IP mobility and SIP multimedia communications sessions. A AA-SBC device (SIP proxy), operating as a Diameter client, sends an accounting request to the Diameter server where the Diameter server returns an accounting response to the Diameter client indicating that it has received and processed the accounting request.

Like RADIUS, a Diameter group is a uniquely named object that defines the authentication and accounting services associated with a group of Diameter servers. Including a Diameter group in one or more configurations allows AA-SBC (the Diameter client) to perform user authentication and forward SIP call detail records to Diameter servers.You can create as many unique Diameter groups as you need.

Within a Diameter group, you set the Diameter authentication and accounting modes that you are using, and whether the Diameter group is to be included as a default authentication and accounting group for SIP traffic that is not governed by configured authentication and accounting policies.

Diameter is required for intercluster route-server. See the master service description for configuration instructions.

AA-SBC can function as either a Diameter client or server.

Note: The Diameter client and server objects are located in different places in the CLI hierarchy. You configure the client within the VSP object and the server on an IP interface.

As a client, it forwards requests for authentication and LCR lookup but does not, in itself, act as a server in accepting requests. As a server, it accepts and responds to requests. When you configure the client (which points to processing servers), you must be sure to configure the server (interface) on the target as well.

Diameter group (client) object summary

The following table lists and briefly describes the **diameter-group** objects. See the following chapter for other objects in the CLI hierarchy:

- Chapter 75, "Virtual system partition (VSP) objects"

Object name	Description
`diameter-group`	Sets general Diameter configuration parameters that will apply to all servers within the group.
`server`	Configures Diameter server parameters within the group.

Diameter interface (server) object summary

The following table lists and briefly describes the **diameter** objects. See the following chapter for other objects in the CLI hierarchy:

- Chapter 14, "Cluster, box, and interface objects"
- Chapter 77, "VLAN objects"
- Chapter 35, "IP objects"

Object name	Description
`diameter-group`	Configures AA-SBC to behave as a Diameter server on the current IP interface.
`port`	Configures interface port parameters for Diameter.

diameter-group

Purpose

Configures a Diameter group, to which you add servers using the server object. Specify a new or existing Diameter group name using up to 16 alphanumeric characters with no blank spaces.

Syntax

```
config vsp diameter-group groupName
```

Properties

Property name	**Description**
admin {enabled | disabled}	Enables or disables the AA-SBC Diameter group configuration. When enabled, the system forwards SIP call detail records and Diameter requests to configured Diameter group server(s). **Example: `set admin enabled`** The default setting is **enabled**.
authentication-mode {round-robin | fail-over *retries* | prioritized}	Sets the Diameter group authentication operational algorithm. • **round-robin**—If you configure multiple authentication servers in the Diameter group, the round robin algorithm performs continued authentication requests to primary and secondary servers until a valid authentication response is received. • **fail-over**—If you configure multiple authentication peer servers in the Diameter group, the failover algorithm forwards authentication requests to secondary servers should the current authentication server fail the number of times specified. • **prioritized**—the system forwards authentication requests to the server with the highest assigned priority. If that server does not respond, the system forwards the request to the next highest priority server. Set the priority with the server **priority** property. **Example: `set authentication-mode round-robin`** The default setting is **fail-over** with **3** retries.

Property name	Description
application {SIP | 3GPPRx | Routing}	Specifies the application that the servers in this Diameter group will be used for. Choose **SIP** for standard AAA activities, **3GPPRx** for interoperation with the Camiant policy server (enabled with the Rx object), and **Routing** for route-server between clusters. **Example: `set application Routing`** The default setting is **SIP**.
origin-host string	Specifies the text that AA-SBC writes to the Origin-Host attribute field in any Diameter requests it sends. This should be the DNS name of the system that you are configuring. **Example: `set origin-host east.cov.com`** There is no default value.
origin-realm *string*	Specifies the text that AA-SBC writes to the Origin-Realm attribute field in any Diameter requests it sends. This should be the domain name of the system that you are configuring. **Example: set origin-realm cov.com** There is no default value.
default-destination-realm string	Specifies the text that AA-SBC writes to the Destination-Realm attribute field in any Diameter responses it sends. This should be the realm of the box you are connecting to, and is for use with the 3Gpp Rx application. For example, if you are connecting to a system with an origin-realm of cov.com, this property must be set to cov.com as well. **Example: `set default-destination-realm cov.com`** There is no default value.

server

Purpose

Identifies and configures the Diameter servers that are part of this Diameter group. Enter a host name of IP address to identify the server.

Syntax

```
config vsp diameter group groupName server serverName
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables the system Diameter authentication and accounting peer server configuration. When enabled, authentication and SIP call accounting records are forwarded to the Diameter peers. **Example: `set admin enabled`** The default setting is **enabled**.
port *portNumber*	Sets the TCP port number over which the AA-SBC client sends authentication requests to the Diameter peer server. **Example: `set port 3888`** Enter a value in the range of 1 to 65535; the default setting is TCP port **3868**.
transport {None | tcp | tls | sctp)	Specifies the IP protocol to use for transmitting Diameter protocol requests and responses between AA-SBC and the Diameter peer server. **Example: `set transport tls`** The default setting is **tcp**.
tls-certificate *certReference*	Assigns the certificate that this Diameter server and its partners present if the protocol is set to TLS. Enter a reference to a previously configured certificate, used by all members of this partnership. **Example: `set tls-certificate “vsp tls certificate diameterCert”`** There is no default setting.

Property name	**Description**
request-timeout *seconds*	Specifies the time (in seconds) to elapse before a request to a Diameter server times out. At that point, TCP retries the request. **Example: `set request-timeout 2`** Enter a value from 1 to 65535; the default setting is **1** second.
window *integer*	Sets the maximum number of simultaneous requests the client can send to the Diameter server. **Example: `set window 6`** Enter a value from 1 to 127; the default setting is **8** requests.
priority *integer*	Configures a priority for the server. Set this property if the **authentication-mode** property of the diameter-group object is set to **prioritized**. The lower the value, the higher the priority. Note that each server in a Diameter group must have a different priority for prioritization to work correctly. **Example: set priority 10** Enter a value between 1 and 99. The default setting is 1.

diameter

Purpose

Identifies the IP interface on which the Diameter server application resides. This is the interface on AA-SBC that listens for incoming Diameter connections. This interface must be configured on each AA-SBC device that is pointed to by a server in a Diameter group.

Syntax

```
config cluster box number interface ethX ip name diameter
config cluster box number interface ethX vlan number ip name diameter
config box interface ethX ip name diameter
config box interface ethX vlan number ip name diameter
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables the system Diameter authentication and accounting peer server configuration. When enabled, authentication and SIP call accounting records are forwarded to the Diameter peers. **Example: `set admin enabled`** The default setting is **enabled**.
origin-host *string*	Specifies the text that AA-SBC writes to the Origin-Host attribute field in any Diameter responses it sends. This should be the DNS name of the system that you are configuring. **Example: `set origin-host east.cov.com`** There is no default value.
origin-realm *string*	Specifies the text that AA-SBC writes to the Origin-Realm attribute field in any Diameter responses it sends. This should be the domain name of the system you are configuring. **Example: `set origin-realm cov.com`** There is no default value.

port

Purpose

Specifies properties for incoming connections. If these settings are not met, the Diameter negotiation fails, and AA-SBC cannot use that connection. Enter a port number that serves as the listener port on this IP interface.

Syntax

```
config cluster box number interface ethX ip name diameter port
   portNumber
config cluster box number interface ethX vlan number ip name diameter
   port portNumber
config box interface ethX ip name diameter port portNumber
config box interface ethX vlan number ip name diameter port portNumber
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables this property profile for incoming connections. **Example: `set admin enabled`** The default setting is **enabled**.
transport {None | tcp | tls | sctp}	Specifies the protocol that the incoming connection must use to communicate with AA-SBC on this port. **Example: `set transport tls`** The default setting is **tcp**.
application {SIP | 3GPPRx | Routing}	Sets the application that the incoming connection must be running to use this port. **Example: `set application Routing`** The default setting is **SIP**.
tls-certificate *certReference*	Assigns the certificate that the incoming connection must present if the protocol is set to TLS. Enter a reference to a previously configured certificate, used by both the connection and the port. **Example: `set tls-certificate “vsp tls certificate diameterCert”`** There is no default setting.

Property name	Description
peer-access-control {none | transport | host-ip-avp | both}	Specifies how AA-SBC controls incoming peer connections. For all selections other than **none**, the list of allowed peers is specified with the **peer** property. Select one of the following: • **none**—allows incoming connections from all peers. • **transport**—allows incoming connections only if the peer's address is in the list of allowed peers. • **host-ip-avp**—allows incoming connections only if the value in the received Host-IP-Address AVP is in the list of allowed peers. • **both**—allows incoming connections only if the both the peer's address and the value in the received Host-IP-Address AVP are in the list of allowed peers. **Example:** **`set peer-access-control none`** The default setting is **host-ip-avp**.
peer *ipAddress*	Specifies the list of peers that are allowed to connect to this port. This property is not applied if the peer-access-control property is set to **none**. **Example:** **`set peer 192.10.0.10`** There is no default setting.

23. Directory objects

Directory description

Directory naming services are the implementation of a centralized system that automates network management of user data and enables interoperation with other enterprise services. The directory service is a database of user information—data such as name, group membership, address, position, office location, contact information, and any number of other identifiers. Enterprise services work by establishing a directory service that client programs access to look up user entries. From this identification, AA-SBC can apply specific policy actions to users.

AA-SBC supports the following enterprise services:

- Microsoft Active Directory
- Standard-based Lightweight Directory Access Protocol (LDAP)
- IBM Domino Enterprise Directory for Lotus Notes
- Derived directories from XML, CSV, database, and static entries.

Note: While you can configure directory services at any time, you must enable the master-services directory object and select a host box for AA-SBC to use the service. See Chapter 39, "Master services objects" for more information.

Active Directory description

Active Directory is the directory service included with Windows 2000 Server. It identifies all resources on a network, making the information available to appropriately configured users. In addition, it provides security for network objects by verifying identities and controlling access.

By setting the configurable objects of the active-directory container, you are providing AA-SBC with access to the Active Directory service. From here, AA-SBC can access the required databases to derive the recognized SIP addresses within your enterprise.

LDAP description

Lightweight Directory Access Protocol (LDAP) is a protocol definition for accessing specialized databases (directories). LDAP can interact with a variety of databases, and unifies the information for consistent management and security. It allows users to query and update information in an LDAP-based directory service.

For example, email programs use LDAP to look up contact information from a server. It is a standard that provides a centralized, up-to-date phone book that any LDAP-aware client has access to. Data entered into an LDAP server is indexed so that you can retrieve specific entries based on filtering criteria. LDAP uses permissions to set access to the database and/or to specific data within it.

Configure the LDAP server on AA-SBC if you do not use a Windows server in your enterprise. This configuration sets how AA-SBC is to recognize and query the schema. To define filters for LDAP queries, refer to *RFC 2254*, *The String Representation of LDAP Search Filter*s. Also see *RFC 3377*, *Lightweight Directory Access Protocol (v3): Technical Specification*.

Notes directory description

The notes directory is the LDAP directory service used by IBM Lotus Instant Messaging and Web Conferencing (Sametime). Sametime uses the Notes Enterprise Server for both messaging and applications. Notes services provide directory, storage, and web server support to enable synchronous collaboration support for users.

By setting the configurable objects of the notes-directory container, you are providing AA-SBC with access to the Domino Enterprise Directory service. From here, AA-SBC can access the required databases to derive the recognized SIP addresses within your enterprise.

Static directory description

The static directory is a list of users manually entered. Use this directory service if you do not have your users previously entered in a format that AA-SBC can then extract them from (CSV, XML, or a database).

XML directory description

The XML directory is a list of users derived from content of an XML document. Use this directory service if you do not have your users registered in an LDAP directory, but can extract them from an XML file.

Database directory description

The database directory is a directory of users drawn from a series of database tables. Use this directory service if you have your users listed in a database table instead of registered in an LDAP directory.

CSV directory description

The CSV directory is a directory of users derived from a comma separated values (CSV) file. Use this directory service if you do not have your users registered in an LDAP directory, but are able to access them through a CSV file.

CLI hierarchy information

See the following chapters for other objects in the CLI hierarchy:

- Chapter 75, “Virtual system partition (VSP) objects”
- Chapter 27, “Enterprise objects”

Directory object summary

The following table lists and briefly describes the objects of the directory names services. In addition, it indicates whether a given subobject is applicable to a name service directory. Assume each object is applicable to all directory services unless noted otherwise in the description.

Object name	Description
directory	Opens the named directory configuration object.
group-filter	Configures the criteria and action for filtering based on the group attribute. Configures the criteria and action for filtering users in the containing group.
user-filter	Configures the criteria and action for filtering based on the user attribute. Configures the criteria and action for filtering users in the containing group.
group-attributes	Configures which attributes to recognize for use with the group-filter object.
user-attributes	Configures which attributes to recognize for use with the user-filter object and to store in the database.
group	Adds users of the directory service to a named group.
sip-address	Constructs a SIP address using attributes pulled from the LDAP schema. • Active Directory • Notes
group-settings	Configures the query into the LDAP schema to return a list of groups. • LDAP
user-settings	Configures the query into the LDAP schema to return a list of users. • LDAP
alias	Configures the translation by which phone numbers are pulled from the LDAP schema.
authentication	Configures generic LDAP settings for the system to use in authenticating users. • LDAP

Object name	Description
user	Creates or modifies a user entry in a static directory. • Static
attribute	Sets the mechanism with which to pull user attributes into the directory service. • XML
credentials	Sets the mechanism with which to incorporate authentication credentials. • XML

directory

Purpose

Opens the directory gateway configuration object to allow setting the parameters for communication between the directory server and AA-SBC, supporting the following enterprise directory services:

- Microsoft Active Directory (active-directory)
- Standard-based Lightweight Directory Access Protocol (ldap)
- IBM Domino Enterprise Directory for Lotus Notes (notes-directory)
- Derived directories from XML, CSV, database, and static entries

Specify the name of the configuration instance that you want to open. If the name does not already exist, the system creates (and opens) an instance by that name. If the name does already exist, the system opens that instance.

Note: While you can configure directory services at any time, you must enable the master-services directory object for AA-SBC to use the service. See Chapter 39, “Master services objects” for more information. In addition, you must set the **local-directory-based-user-services** property of the settings object to **enabled** to perform directory-based user services for SIP traffic.

Using XPath Queries

XPath is a language used to address and extract information from an XML document. The XML directory allows you to use XPath queries to derive the users you want extracted for the enterprise name service. For more information on XPath queries, see the *XML Path Language (XPath) W3C Recommendation.*

The properties listed in the following table describe all possible properties for directories. However, each directory has a unique selection of parameters. The following table summarizes the property parameters per directory service. Those properties marked are available in the corresponding directory.

Property name	Active Directory	LDAP	Notes Directory	Static	XML	Data base	CSV
admin	√	√	√	√	√	√	√
tag	√	√	√	√	√	√	√
domain	√	√	√	√	√	√	√
default-policy	√	√	√	√	√	√	√
to-policy	√	√	√	√	√	√	√
from-policy	√	√	√	√	√	√	√
user-group-policy	√	√	√	√	√	√	√
host	√	√	√				
port	√	√	√				
transport	√	√	√				
timeout	√	√	√				
username	√	√	√			√	
password-tag	√	√	√			√	
organization			√				
id-attribute	√		√				
ignore-unresolved	√	√	√				
primary-group	√						
ignore-domain	√	√	√				
group-id-map		√					
use-aliases	√						
parent-directory							

Property name	Active Directory	LDAP	Notes Directory	Static	XML	Data base	CSV
source					√		√
user-query					√		
name-subquery					√		
address-subquery					√		
group-subquery					√		
connection						√	
query						√	
column						√	√
has-header-row							√

In addition, each description notes any directory it is not affiliated with. (Assume each property is applicable to all directory services unless noted otherwise in the description.)

Syntax

```
config vsp enterprise directories active-directory string
config vsp enterprise directories ldap string
config vsp enterprise directories notes-directory string
config vsp enterprise directories static-directory string
config vsp enterprise directories xml-directory string
config vsp enterprise directories database-directory string
config vsp enterprise directories csv-directory string
```

Properties

Property name	Description
admin {enabled | disabled}	Specifies whether AA-SBC uses this directory configuration in the current session. **Example: set admin enabled** The default setting is **enabled**.
tag *string*	Defines a tag to attach to a SIP URI, indicating where the information was learned from. You can use this tag as an identifier in policy configuration. **Example: set tag fromNotes** There is no default setting.
domain *domainName*	Sets the domain name, which is the root search context for locating users and groups within that domain. If the username property does not contain a domain name in it, this domain is attached. Enter either the real domain that the directory is servicing or enter an alias domain, which is a domain name that AA-SBC associates with the default domain. The system stores only a single domain property. If you reissue the command, AA-SBC overwrites the previous value. **Example: set domain companyABC.com** There is no default setting.
default-policy *policyReference*	Sets the default policy to apply to all users of this directory. This policy acts as a baseline policy in effect at all times unless more specific policy rules override the default settings. Enter the path to a previously configured policy. **Example: set default-policy “vsp policies session-policies policy notes”** There is no default setting.
to-policy *policyReference*	Specifies the policy to apply to traffic going to users of this directory. **Example: set to-policy “vsp policies session-policies policy incoming”** There is no default setting.

Property name	Description
from-policy *policyReference*	Specifies the policy to apply to traffic coming from users of this directory. **Example: set from-policy “vsp policies session-policies policy outgoing”** There is no default setting.
user-group-policy *groupname policyReference*	Specifies the policy to apply to users of this directory service who are members of the specified group. The group can be either a user group from the directory service schema or a virtual group constructed in the configuration for policy application purposes. Enter a group name, and AA-SBC applies the specified policy to any user belonging to that group. Also enter the complete path to a previously configured policy reference. **Example: set user-group-policy grp3 “vsp policies session-policies policy notesGroup”** There is no default setting.
host *name*	Sets the identifier of the directory host server. Specify the either an IP address in standard dotted decimal format or specify a network-recognized host name. **Example: set host 10.10.5.1** There is no default setting. • active-directory • ldap • notes
port *portNumber*	Sets the known port number used by this directory service. **Example: set port 1010** Enter a port number between 1 and 65535. The default setting is **389**. • active-directory • ldap • notes

Property name	Description
transport {tcp | tls *certificateReference*}	Sets the connection type to the LDAP server, either secure or not. For a secure connection, set transport to TLS and include a reference to a certificate on the system. **Example: set transport tls “vsp tls certificate nnos-e.abc.com”** The default setting is **tcp**. • active-directory • ldap • notes
timeout *milliseconds*	Sets the number of milliseconds that the system attempts to contact this directory server. When the timeout expires, the action is determined by the enterprise-level **on-failure** setting. **Example: set timeout 5000** Enter a value between 1 and 65535. The default setting is **15000** ms (15 seconds). • active-directory • ldap • notes
username *string*	Configures the name of the person qualified to log into this directory server. This name must match the username configured on the server. **Example: set username admin** There is no default setting. • active-directory • ldap • notes • database-directory

Property name	Description
password-tag *string*	Specifies the tag associated with the shared secret used to authenticate transactions between AA-SBC and this directory server. See Using passwords and tags for information on the AA-SBC two-part password mechanism. **Example: set password-tag admin** There is no default setting. • active-directory • ldap • notes • database-directory
organization *string*	Specifies a grouping attribute in the Notes directory LDAP schema. AA-SBC can use this attribute to identify users from that server. **Example: set organization HR** There is no default setting. • notes
id-attribute *string*	Specifies an attribute defined in this directory schema that AA-SBC uses to identify users from that server. **Example: set id-attribute ADaccount** The default attribute is **samaccountname**. • active-directory • notes

Property name	Description
ignore-unresolved {true | false}	Sets how the system handles unresolved SIP addresses. These are lookups that resulted in an error message indicating that the user was not found. Select either: • **true**—sets the system to ignore the message, but to send a notification of the failure to the event log. • **false**—sets the system to fail the action, and to revert to the enterprise-level **on-failure** setting to determine the next action. **Example: set ignore-unresolved false** The default setting is **true**. • active-directory • ldap • notes
primary-group `id# groupName`	Maps the primaryGroupID number to a valid group name. If you do not know the group name, you must run a discovery tool, such as the one found at http://www.jsiinc.com/SUBP/tip7700/rh7729.htm, to determine the name of the group that maps to the ID number. Identify the group by: • *id#*—specifies the number associated with the primaryGroupID field in the Active Directory schema. • *string*—enter a valid group name from the Active Directory schema. **Example: set primary-group 513 domainUsers** There is no default setting. • active-directory

Property name	Description
ignore-domain {true | false}	Sets whether the domain portion of SIP addresses are significant. When AA-SBC receives a request (which has a domain tagged to the SIP address), this setting controls whether the system makes use of the domain. Select either: • **true**—the system ignores the domain. Use this setting, for example, if the SIP address mapping is virtualized. In this case you would want the system to forward based on the user tag. • **false**—the system acknowledges the domain and acts according to settings pertaining to domain tags. **Example: set ignore-domain false** The default setting is **true**. • active-directory • ldap • notes
group-id-map *id group*	Maps a number to the name of a group. This field may be required in instances where the LDAP schema uses a numerical ID instead of a name. Use this property in conjunction with the **group-id-attribute** property in user-settings to indicate which attribute in LDAP you are using to identify users. For example, set this property to map engineering to ID 2, and set the **group-id-attribute** property to GID. The GID attribute contains the value 2, which would be mapped to the group named “engineering.” Enter an ID number and the name of an existing LDAP group. If the specified name does not exist on the LDAP server, the mapping is ignored. **Example: set group-id-map 2 engineering** There is no default setting. • ldap

Property name	Description
use-aliases {true | false}	Sets whether AA-SBC can use a proxy alias to determine the SIP address. Select either: • **true**—the system uses proxy aliases. For example, an SMTP proxy address can be used in place of the SIP address. • **false**—the system only uses configured SIP addresses. **Example: set use-aliases false** The default setting is **true**. • active-directory
source *`path`*	Specifies the location of the document that contains either the XML data or CSV file. AA-SBC supports accessing the files though FTP or HTTP, as well as pointing to a local file. **Example:** **`set source ftp://myftp.companyABC.com/users.xml`** **`set source http://www.companyABC.com/users.csv`** **`set source file:/cxc/users.xml`** There is no default setting. • xml-directory • csv-directory
user-query *`string`*	Configures an XPath query that returns all nodes that are user nodes. Specify the string that identifies your user nodes. **Example: set user-query //user** There is no default setting. • xml-directory

Property name	Description
name-subquery *string*	Configures an XPath query that returns all user names. The query is executed based on the user node, as identified in the user-query property. **Example: set name-subquery /user/@name** There is no default setting. • xml-directory
address-subquery *string*	Configures an XPath query that returns, for each user, all SIP addresses. The query is executed based on the user node, as identified in the user-query property. **Example: set address-subquery /user/address/text()** There is no default setting. • xml-directory
group-subquery *string*	Configures an XPath query that returns, for each user, all group name(s). The query is executed based on the user node, as identified in the user-query property. **Example: set group-subquery /user/@group** There is no default setting. • xml-directory
connection {oracle *IPaddress* [sid] | postgres *IPaddress* [*databaseName*] | sqlserver *IPaddress* [*databaseName*] | generic *url driver validation*}	Specifies the information necessary for AA-SBC to connect to the database containing the user data. **Example: set connection oracle 192.168.215.222:1521 db** There is no default setting. • database-directory
query *string*	Configures a query that the system uses to extract users from the selected database. **Example: set query “select * from users”** There is no default setting. • database-directory

Property name	Description
column {name *number* | address *number* | group *number* | attribute *number* name | principal *number* | password *number* | uri *number*}	Specifies the position each specific field should take. Re-execute the command for each column you want to map to. **Example: `set column name 1`** **`set column address 2`** **`set column group 3`** **`set column group 4`** **`set column attribute 5 ipphone`** There is no default setting. • csv-directory • database-directory
has-header-row {false | true}	Indicates to the system whether there is a header row in the data that should be ignored on import. **Example: set has-header-row true** The default setting is **false**. • csv-directory

`group-filter`

Purpose

Sets the filter criteria by which to include or exclude groups from AA-SBC. This is the filter by which you import users, based on their group associations, from the LDAP server. Any excluded groups (and their member users) are not considered valid by the system. The results of this filter are then acted on by the filters set in the group object filter.

Group filter works on two levels—as a subobject of the named directory or as a subobject of the group object.

As a subobject of a named directory (vsp\enerprise\directories\directory\group-filter), the filter returns data from the LDAP server on users of those groups matching the filter criteria. You can then create a virtual group, using the group object, by further filtering members of the group(s) remaining from any previous filter process. This is done with the second instance of the group-filter object, found in, for example, vsp\enerprise\directories\directory\group\group-filter.

Syntax

```
config vsp enterprise directories directory name group-filter
config vsp enterprise directories directory name group name
   group-filter
```

Properties

Property name	Description
type {include | exclude}	Sets whether to include or exclude groups matching the name and/or attribute criteria (below) specified. When configuring a group filter, you must specify this property. Select either: • **include**—allows all groups matching the criteria to be included in the system records. • **exclude**—prohibits all groups matching the criteria from being included in the system records. **Example: set type include** The default setting is **include**.

Property name	Description
name {matches *regExp* | equals *string*}	Sets a string to match the name of the group against. Any groups meeting the name criteria are then included or excluded, according to the **type** property. Set the string to: • **matches**—match a regular expression that you specify. Enter a regular expression pattern, and all group names that match those characters are acted on. • **equals**—match the entered string exactly. **Example: set name matches ne.*** There is no default setting.
attribute {matches *attribute regExp* | equals *attribute string* | exists *attribute*}	Sets one or more attributes to match the group against. Any groups containing the specified attribute(s) are then included or excluded, according to the **type** property. Re-execute the command to add additional attributes; they are AND'd if there are multiple attribute match criteria. Set the string to: • **matches**—match an attribute that you specify. Enter an attribute name and a regular expression pattern. All group names that have an attribute that matches those characters are acted on. • **equals**—match the entered attribute name exactly. The value retrieved must match the value specified for the attribute exactly. • **exists**—match any user that contains the specified attribute. Enter the full name of the attribute to match against. However, if that name exists as a portion of another name, it will still be returned. For example, “admin” would return everyone containing “admin” and “admin rw”. **Example: set attribute exists admin** There is no default setting.

user-filter

Purpose

Sets the filter criteria by which to include or exclude users from AA-SBC. Any excluded users are not considered valid by the system.

User filter works on two levels—as a subobject of the named directory or as a subobject of the group object.

As a subobject of a named directory (for example, vsp\enerprise\directories\directory\user-filter), the filter returns data from the LDAP server on those users matching the filter criteria. You can then create a virtual group, using the group object, by further filtering remaining users from any previous filter process. This is done with the second instance of the user-filter object, found in vsp\enerprise\directories\directory\group\user-filter.

Syntax

```
config vsp enterprise directories directory name user-filter
config vsp enterprise directories directory name group name
   user-filter
```

Properties

Property name	Description
type {include | exclude}	Sets whether to include or exclude users matching the name and/or attribute criteria specified. When configuring a user filter, you must specify this property. Select either: • **include**—allows all users matching the criteria to be included in the system records. • **exclude**—prohibits all users matching the criteria from being included in the system records. **Example: set type include** There is no default setting.

Property name	Description
name {matches *regExp* | equals *string*}	Sets a string to match the name of the user against. Any users meeting the name criteria are then included or excluded, according to the `type` property. Set the string to: • **matches**—match a string that you specify. Enter a regular expression pattern, and all user names that match those characters are acted on. • **equals**—match the entered string exactly. **Example: set name equals Gaylord** There is no default setting.
attribute {matches *attribute regExp* | equals *attribute string* | exists *attribute*}	Sets one or more attributes to match the user against. Any users containing the specified attribute(s) are then included or excluded, according to the **type** property. Re-execute the command to add additional attributes; they are AND'd if there are multiple attribute match criteria. Set the string to: • **matches**—match an attribute that you specify. Enter an attribute name and a regular expression pattern to compare the value for that attribute. All users that have a matching attributed are acted on. • **equals**—match the entered attribute name exactly. The value retrieved must match the value specified for the attribute exactly. • **exists**—match any user that contains the specified attribute. Enter the full name of the attribute to match against. However, if that name exists as a portion of another name, it will still be returned. For example, “admin” would return everyone containing “admin” and “admin rw”. **Example: set attribute matches phone /d/d/d-/d/d/d-/d/d/d/d** There is no default setting.

group-attributes

Purpose

Sets which additional attributes to retrieve from the records of groups that matched the group-filter criteria.

Syntax

```
config vsp enterprise directories directory name group-attributes
```

Properties

Property name	Description
name *string*	Sets the name of the attribute(s) that should be retrieved and stored with the group record. Re-execute the command for each attribute you want retrieved. **Example: set name department** There is no default setting.

user-attributes

Purpose

Sets which additional attributes to retrieve from the records of groups that matched the user-filter criteria.

Syntax

```
config vsp enterprise directories directory name user-attributes
```

Properties

Property name	Description
name *string*	Sets the name of the attribute(s) that should be retrieved and stored with the user record. Re-execute the command for each attribute you want retrieved. **Example: set name telephoneNumber** There is no default setting.

group

Purpose

Creates a virtual collection of users, derived as a result of filtering the directory LDAP schema. Once the schema has been initially filtered to create a “first pass” of users, you can further refine your membership into a virtual group by opening this object and executing the group-filter object from within it. To explicitly add users, set the **user** property of this command.

Specify the name of the group instance that you want to open. If the name does not already exist, the system creates (and opens) an instance by that name. If the name does already exist, the system opens that instance.

Syntax

```
config vsp enterprise directories directory name group name
```

Properties

Property name	Description
user *string*	Configures the named user as part of the group. Enter a user that is currently registered in the directory service. **Example: set user jdoe** There is no default setting.

sip-address

Purpose

Configures the mechanism by which to derive the user SIP address from the attributes pulled from the LDAP schema. You can construct a user SIP address using either any corresponding piece of the user data in the schema or your own entries. The resulting string is the complete string AA-SBC uses as the user SIP address.

Only applicable to

- Active Directory
- Notes

Syntax

```
config vsp enterprise directories directory name sip-address
```

Properties

Property name	Description
value *attribute*	Defines the attribute to use from the directory schema. Use tokens to identify the attribute name that will be recomposed into the SIP address. **Example: set value %userID%** There is no default setting.
pattern *`regExp composition`*	Configures AA-SBC to use only a part of the specified attribute name. If you do not specify this property, the system uses the entire attribute extracted with the `value` property. To use this property, specify: • *regExp*—enter a regular expression identifying the portion of the attribute to be added to the record. For example, the following expression identifies a subexpression (between the parenthesis): .*(\d\d\d\d)$ • *composition*—enter a string that defines how to recompose the resulting regExp string. The resulting string is stored for that user in the system database. In the following example, the first component from the regular expression is substituted in place of the “1” and appended to the “x.” The percent signs identify the variable to be replaced: x%1% **Example: set pattern .*(\d\d\d\d)$ x%1%** There is no default setting.

group-settings

Purpose

Configures the criteria by which you query the LDAP schema to return a list of groups for use by AA-SBC.This object is only relevant to LDAP and Active Directory (AD) configurations. Most properties are for the LDAP directory; AD learns values for those fields dynamically.

➡ **Note:** You must be familiar with your LDAP schema to configure group settings, as the properties require that you set attribute names. In addition, be certain to refer to the LDAP RFCs for syntax entry requirements.

Notes on queries vs. attributes

The properties that you can set with this object include, in some cases, both queries and attributes as identifiers for extracting users. For example, this object has a **member-of-attribute** and a **member-of-query** property.

Configuring an attribute instructs AA-SBC to extract all users sharing that attribute. For example, setting member-of-attribute to admin returns all users that are a member of the group “admin.” The query returns, in the member-of case, a list of all groups matching it.

An attribute and query of the same type are mutually exclusive because you can only configure AA-SBC to extract users from one source. So, AA-SBC can extract on the attribute you specify or on the results of the query you configure.

Writing derived token strings

You can use token expansion when configuring queries. In addition, queries use logical AND, OR, and NOT. Keep in mind the following rules when writing strings:

- No special characters indicate literal text.
- The percent sign (%) delineates a variable pulled from the original user record.

No preceding special character within the variable statement indicates a field name, for example, ID, name, extension, SIP-address, etc.

The at sign (@) indicates that an attribute name immediately follows.

- If the field is a vector (list) you must supply brackets to indicate which entry to use. For example, alias[3] specifies to use the third alias domain name configured in the record.

For example:

```
SIP:%@telephoneNumber%@companyABC.com
```

might translate to:

```
SIP:555-1212@companyABC.com
```

Only applicable to

- LDAP
- Active Directory

Syntax

```
config vsp enterprise directories ldap name group-settings
```

Properties

Property name	Description
base *string*	Sets the point in the LDAP directory structure at which the system should start its queries for groups. **Example: set base cn=Servers** The default base is **cn=Users**.
Active Directory only second-base *string*	Sets a secondary point in the LDAP directory structure at which the system should start its queries for groups. This search structure is used in addition to the initial base search. **Example: set second-base cn=People** There is no default secondary base.
LDAP only use-subtree {true | false}	Determines whether the system searches all subtrees whose root is the provided base. Set to **true** to query all branches; set to **false** to query only the root. **Example: set use-subtree false** The default setting is **true**.

Property name	Description
query *string*	Defines how to get a list of groups that match the query criteria. **Example: set query (&(objectClass=InetOrgPerson))** The default setting is **(&(objectClass=groupOfNames))**.
LDAP only id-attribute *string*	Sets the ID attribute that a user must match to be returned on the query. **Example: set id-attribute purge** The default ID attribute setting is **cn** (common name).
LDAP only name-attribute *string*	Sets the name attribute that a user must match to be returned on the query. **Example: set name-attribute sn** The default setting is **cn** (common name).
LDAP only description-attribute *string*	Sets the description attribute that a group must match to be returned on the query. **Example: set description-attribute male** The default setting is **description**.
LDAP only `member-attribute` *`string`* `{dn | name}`	Sets the attribute name used to obtain a list of all groups containing this attribute. AA-SBC returns all groups containing this attribute. Also, specify whether the string specified is a name or domain name (dn). You cannot set both this property and the **member-query**. **Example: set member-attribute east** There is no default setting for the string. The default type is **dn**.
LDAP only `member-query` *`string`*	Returns a list of groups that contain the specified member attribute. You cannot set both this property and the **member-attribute**. **Example: set member-query admin** There is no default setting.

Property name	Description
LDAP only `member-of-attribute` *`string`* `{dn | name}`	Sets the name of the attribute used to obtain a list of all groups containing this attribute. Also, specify whether the string specified is a name or domain name (dn). You cannot set both this property and the **member-of-query**. **Example: set member-of-attribute memberOf** There is no default setting for the string. The default type in **dn**.
LDAP only `member-of-query` *`string`*	Returns a list of users that are members of the specified group(s). You cannot set both this property and the **member-of-attribute**. **Example: set member-of-query admin** There is no default setting.
LDAP only `object-class` *`string`*	Sets the object class that a group must belong to in order to be returned as a member group. The values of the objectClass attribute determine the schema rules the entry must obey. **Example: set object-class sales** The default setting is **groupOfNames**.

user-settings

Purpose

Configures the criteria by which you query the LDAP schema to return a list of users for use by AA-SBC. This object is only relevant to LDAP and Active Directory (AD) configurations. Most properties are for the LDAP directory; AD learns values for those fields dynamically.

➡ **Note:** You must be familiar with your LDAP schema to configure user settings, as the properties require that you set attribute names. In addition, be certain to refer to the LDAP RFCs for syntax entry requirements.

See Notes on queries vs. attributes for more information on the distinction between queries and attributes.

Only applicable to

- LDAP
- Active Directory

Syntax

```
config vsp enterprise directories ldap name user-settings
```

Properties

Property name	Description
base *string*	Sets the point in the LDAP directory structure at which AA-SBC should start its queries for users. **Example: set base ou=Servers** The default base is **ou=People**.
LDAP only use-subtree {true | false}	Determines whether AA-SBC searches all subtrees whose root is the provided base. Set to **true** to query all branches; set to **false** to query only the root. **Example: set use-subtree false** The default setting is **true**.
query *string*	Defines how to get a list of users that match the query criteria. **Example: set query (&(objectClass=InetOrgPerson))** The default setting is **(&(objectClass=person))**.
LDAP only id-attribute *string*	Sets the ID attribute that a user must match to be returned on the query. **Example: set id-attribute purge** The default ID attribute setting is **uid** (user ID).
LDAP only name-attribute *string*	Sets the name attribute that a user must match to be returned on the query. **Example: set name-attribute sn** The default setting is **cn** (common name).
LDAP only description-attribute *string*	Sets the description attribute that a user must match to be returned on the query. **Example: set description-attribute male** The default setting is **description**.

Property name	Description
LDAP only `group-id-attribute` *`string`*	Sets the name of the attribute in the user object that contains the users group ID, if one exists. AA-SBC matches the group ID against the **group-id-map** property of the **ldap** user-settings object to determine the group name. For example, set the **group-id-map** property to map engineering to ID 2, and set this property to GID. The GID attribute contains the value 2, which would be mapped to the group named “engineering.” **Example: set group-id-attribute GID** There is no default setting.
LDAP only `member-of-attribute` *`string`* `{dn | name}`	Sets the name of the attribute used to obtain a list of all groups to which the user belongs. Also, specify whether the string specified is a name or domain name (dn). You cannot set both this property and the **member-of-query**. **Example: set member-of-attribute memberOf** There is no default setting for the string. The default type in **dn**.
LDAP only `member-of-query` *`string`*	Returns a list of groups that a user is a members of. You cannot set both this property and the **member-of-attribute**. **Example: set member-of-query admin** There is no default setting.
LDAP only `object-class` *`string`*	Sets the object class that a user must belong to in order to be returned as a member user. The values of the objectClass attribute determine the schema rules the entry must obey. **Example: set object-class sales** The default setting is **person**.
LDAP only `sip-address` *`string`*	Configures the way in which to derive the user SIP address from the LDAP schema. **Example: set sip-address sip:** There is no default setting.

alias

Purpose

Configures the extension attributes for a user telephone entry based on the LDAP attribute that contains the telephone description. This object requires that you enter an attribute name to open the object. Enter an attribute name from the LDAP schema.

Syntax

```
config vsp enterprise directories directory name alias attributeName
```

Properties

Property name	Description
pattern *regExp composition*	Configures AA-SBC to use only a part of the specified attribute name. If you do not specify this property, the system uses the entire attribute entered when opening this object. To use this property, specify: • *regExp*—enter a regular expression identifying the portion of the attribute to be added to the record. For example, the following expression identifies a subexpression (between the parenthesis): .*(\d\d\d\d)$ • *composition*—enter a string that defines how to recompose the resulting regExp string. The resulting string is stored for that user in the system database. In the following example, the first component from the regular expression is substituted in place of the “1” and appended to the “x.” The percent signs identify the variable to be replaced. x%1% **Example: set pattern .*(\d\d\d\d)$ x%1%** There is no default setting.

authentication

Purpose

Configures the value that AA-SBC offers in response to an authentication challenge from an LDAP server.

Only applicable to

- LDAP

Syntax

```
config vsp enterprise directories ldap name authentication
```

Properties

Property name	Description
user-pattern *pattern*	Defines the attribute to use from the directory schema. Use tokens to identify the attribute name that will be recomposed into the into the user name offered in response to an authentication challenge. **Example: set authentication %userID%** There default setting is **%id%**.

user

Purpose

Creates or modifies the configuration for an individual user in the static directory. Enter a user name to open the user object, and then use the attribute and credentials subobjects to enter information for the user.

Only applicable to

- Static directory

Syntax

```
config vsp enterprise directories static-directory name user name
```

Properties

Property name	Description
address *sipAddress*	Sets the SIP address of the user. A user can only be associated with a single SIP address. Re-executing the command overwrites the previous value. **Example: set address sip:JDoe@abc.com** There is no default setting.
group *string*	Assigns the user to a group. You can enter any number of group associations. **Example: set group admins** There is no default setting.

attribute

Purpose

Sets the mechanism with which to pull user attributes into the directory service. With static-directory, you input user attributes directly. With xml-directory, you configure queries that return the values for the attributes. In either case, the results are written to the directory service, and provide the necessary information for AA-SBC.

To use the attribute object within static-directory, you must supply an attribute name to open the object.

Only applicable to

- Static directory
- XML directory

Syntax

```
config vsp enterprise directories static-directory name user name
   attribute string
config vsp enterprise directories xml-directory name attribute
```

Properties

Property name	Description
attribute-subquery *string*	Configures an XPath query that returns all attributes having the specified node tag, based on the user node. (User node was specified in the user-query property of the xml-directory object.) Specify the string that identifies your attribute nodes. **Example: set attribute-subquery //attribute** There is no default setting. • xml-directory
name-subquery *string*	Configures an XPath query that returns the names of attributes matching the specified string. The query is executed based on the attribute node, as identified in the attribute-subquery property. **Example: set name-subquery /attribute/@name** There is no default setting. • xml-directory
values-subquery *string*	Configures an XPath query that returns values for the named attributes. The query is executed based on the attribute node, as identified in the attribute-subquery property, and the attribute name from the name subquery. **Example: set values-subquery //values/text()** There is no default setting. • xml-directory
value *string*	Sets the value of the named attribute. You can enter any number of values for an attribute. **Example: set value ALeast** There is no default setting. • static-directory

credentials

Purpose

Configures the mechanism for adding authentication credentials to user records. With static-directory, you input user credentials directly. With xml-directory, you configure queries that return the values for the credentials. In both cases, the results are written to the directory service, and provide the necessary information for AA-SBC to supply authentication information when challenged.

Only applicable to

- Static directory
- XML directory

Syntax

```
config vsp enterprise directories static-directory name user name
   credentials
config vsp enterprise directories xml-directory name credentials
```

Properties

Property name	Description
user-subquery *string*	Configures an XPath query that returns user names. The query is executed based on the user node, as identified in the user-query property of the xml-directory object. **Example: set user-subquery /user/credential/@user** There is no default setting. • xml-directory
password-subquery *string*	Configures an XPath query that returns user passwords. The query is executed based on the user node, as identified in the user-query property of the xml-directory object. **Example: set password-subquery /user/credential/@password** There is no default setting. • xml-directory

Property name	Description
uri-subquery *`string`*	Configures an XPath query that returns replacement URIs. These alternate SIP address are used for authentication. The query is executed based on the user node, as identified in the user-query property of the xml-directory object. **Example: set uri-subquery /user/credential/@uri** There is no default setting. • xml-directory
user *`string`*	Sets the user name to supply for authentication. **Example: set user admin** There is no default setting. • static-directory
password *`string`*	Sets the password to supply for authentication. **Example: set password admin** There is no default setting. • static-directory
uri *`string`*	Sets the replacement URI to be used for authentication. **Example: set uri sip:admin@abc.aleast.com** There is no default setting. • static-directory

24. Display name character set objects

Display name character set description

Phones (user agents) have diverged in the character sets that they support. Older models of firmware may not support the same character sets as newer models. When this happens, the caller ID display is usually left empty. AA-SBC performs transliteration from one character set to another in order to display the content of the From header in the caller ID display.

This object configures the character set translation that AA-SBC performs on display names. When AA-SBC receives a call, it does a lookup on the destination. If the destination phone is in the location cache, AA-SBC compares the User Agent string to the configured transliteration list, and converts display names, where applicable.

Display name object summary

The following table lists and briefly describes the **displayname-character-set-info** object. See the following chapter for other objects in the CLI hierarchy:

- Chapter 75, “Virtual system partition (VSP) objects”

Object name	Description
`displayname-character-set-info`	Specifies the character set that a phone can accept.

displayname-character-set-info

Purpose

Specifies the character set that a phone type can accept. AA-SBC performs a character set conversion on the user agent(s) you specify when transmitting to those user-agents. This change only applies to the display name in the From header.

Syntax

```
config vsp displayname-character-set-info
```

Properties

Property name	Description
user-agent *userAgent* {unchanged | utf-8 | ascii-7bit | iso-8859-1}	Specifies the character set that the system should convert into when the transmitting to the configured user agent. Enter a user agent or regular expression and select the character set. You can only set one character set per user agent entry, but you can distinguish user agents (for example, by release) and set different character sets for each. By specifying the character set that the phone accepts, you set the system to: • **unchanged**—make no modification to the characters. • **utf-8**—use the UTF-8 characters set (representing accented characters with multibyte encoding). • **ascii-7bit**—translate all accented characters into 7-bit ASCII. • **iso-8859-1**—use the ISO 8859-1 character set (representing accented characters with single byte encoding). **Example: set user-agent snom* utf-8** The default setting is **unchanged**.

25. DNS service resolver and server objects

DNS description

Domain Name System (DNS) servers are responsible for translating Internet host names to IP addresses. For example, DNS converts the name entered on a Web browser address bar to the IP address of the Web server that hosts that particular Web site. DNS uses a distributed database to store this name and address information for all public hosts on the Internet.

When an Internet client issues a request that involves an Internet host name, a DNS server determines the host IP address. If the DNS server cannot service the request, it sends the request to other DNS servers until the IP address is resolved, completing the Internet client request.

AA-SBC can service both DNS requests from internal client processes (resolver function) or act as a DNS server. As a resolver residing in the management process, it accepts requests for resolutions from client processes (e.g., SIP, LCR). The resolver maintains a cache of entries as does each client process. As a server, AA-SBC accepts both internal and external queries. It may first try to use the resolver to respond to them or it may immediately forward (proxy) them to an external server. See the dns-server object description for request processing details.

➡ **Note:** The DNS resolver and server objects are located in different places in the CLI hierarchy. You configure the resolver within the VSP object and the server on an IP interface.

Within the **dns** object, you can create static configurations to map a host or a domain to a SIP service. These are maintained in DNS NAPTR and SRV records. AA-SBC also maintains a cache of query responses which it consults for information before querying an external server.

Understanding FQDN and single-label queries

AA-SBC has features in the resolver to process and attempt resolution of single-label queries. A single-label query is just that—a label that is not a fully qualified domain name (FQDN). Depending on the setting of the **send-single-label-query** property, AA-SBC attempts to resolve these queries internally and/or externally. If the name is an FQDN, the system does not append a domain or search name. The following table illustrates resolution for different query types. For this example, assume that the domain-name is set to nnos-e.com and additional search domains are set to search1.com and search2.com.

Query type	Example	Resolutions
Single-label	abc	abc abc.nnos-e.com abc.search1.com abc.search2.com
Two label	abc.com	abc.com abc.com.nnos-e.com abc.com.search1.com abc.com.search2.com
FQDN	abc.com.	abc.com.

For more information on DNS, refer to:

- *RFC 2782, A DNS RR for specifying the location of services (DNS SRV)*
- *FC 2915, The Naming Authority Pointer (NAPTR) DNS Resource Record*

DNS client object summary

The following table lists and briefly describes the **dns** objects. See the following chapter for other objects in the CLI hierarchy:

- Chapter 75, "Virtual system partition (VSP) objects"

Object name	Description
`dns`	Opens the DNS configuration object for editing.
`resolver`	Sets properties of the DNS service resolver function.

Object name	Description
server	Sets properties of the DNS resolver servers.
host	Maps a host name to an IP address for use with DNS lookups.
service	Maps priority information about a service to a server.
naptr	Sets the lookup procedure for destinations with unknown protocols or ports.
reject	Sets the system to always reject specific entries.

DNS server object summary

You configure DNS servers on IP interfaces.

The following table lists and briefly describes the **dns-server** objects. See the following chapters for other objects in the CLI hierarchy:

- Chapter 14, “Cluster, box, and interface objects”
- Chapter 77, “VLAN objects”
- Chapter 35, “IP objects”

Object name	Description
dns-server	Configures the system to behave as a DNS server on the current IP interface.

dns

Purpose

Opens the DNS configuration object for editing. From within this object you configure DNS service and resolver characteristics and identify external servers that can respond to requests not satisfied through the internal DNS cache.

Syntax

```
config vsp dns
```

Properties

None

resolver

Purpose

Sets the characteristics of the AA-SBC resolver function. As a resolver, AA-SBC obtains resource records from servers on behalf of resident or requesting applications.

Syntax

```
config vsp dns resolver
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables the DNS configuration. **Example: set admin enabled** The default setting is **enabled**.
server-scheme {preference-order | load-balance | least-cost}	Specifies how the system selects the server to which it forwards DNS queries. The system selects based on: • **preference-order**—the preference assigned to the server with the **preference** property. • **load-balance**—criteria such as pending queries and number of previous requests. • **least-cost**—the best performance, based on, in part, response time in previous queries. **Example: set server-scheme load-balance** The default setting is **preference-order**.
query-timeout *seconds*	Specifies the time, in seconds, that a lookup can go unanswered before it times out. Once it times out, the system retries for the configured number of times. **Example: set query-timeout 3** Enter a number of seconds between 1 to 10. The default setting is **2 seconds**.
query-retries *retries*	Specifies the number of DNS query (lookup) retries to execute if a DNS query times out. Once the configured number of retries is attempted, the system tries the next configured server. **Example: set query-retries 1** Enter a number of retries between 0 to 10; 0 indicates no retries. The default setting is **2 retries**.
cache-poll-interval *seconds*	Specifies the number of seconds that the system waits between refreshing the DNS cache and cleaning up aged entries. **Example: set cache-poll-interval 120** Enter a number of seconds between 1 to 65535. The default setting is **60 seconds**.

Property name	Description
dead-threshold *queries*	Specifies the number of unanswered queries the system can tolerate before changing the DNS server state to DOWN. The server remains in that state until expiration of the time set with the **dead-interval** property. **Example: set dead-threshold 5** Enter a value between 1 and 65535; the default setting is **10** queries.
not-available-ttl *seconds*	Specifies the number of seconds that the system caches DNS entries that are not found on the server. When the **cache-poll-interval** time expires, the system purges the entry. **Example: set not-available-ttl 4500** The default setting is **3600** seconds.
use-nnos-domain-in-search {enabled | disabled}	Specifies whether the resolver should append the configured domain name to queries it receives that are not FQDNs. If **enabled**, AA-SBC appends the name set with the domain-name property of the static-stack-settings object to queries. **Example: set use-nnos-domain-in-search disabled** The default setting is **enabled**.
additional-search-domains *string*	Specifies additional domains to append to single-label queries. When AA-SBC receives a query which is not an FQDN, it appends the configured domain (if **use-nnos-domain-in-search** is enabled) and any additional domains specified with this property. **Example: set additional-search-domains company.com** There is no default setting.

Property name	Description
enum-domain *string*	Specifies the domain name to append to a phone number when the system performs an ENUM lookup. The ENUM lookup converts a phone number to an IP address. By default, the system uses the standard ENUM DNS domain as specified in *RFC 2916, E.164 number and DNS*. **Example: set enum-domain e164.arpa** The default setting is **e164.arpa**.
dead-interval *seconds*	*Secondary property.* Specifies the number of seconds that a server is considered DOWN by the system. When unanswered responses exceed the threshold set with the **dead-threshold** property, AA-SBC considers the server down. (The server state and the current dead count are reported using the **show dns-resolver -v** command.) The server stays in the down state (and therefore has no queries forwarded to it) for the number of seconds set with this property. When this timer expires, the dead count is reset to zero and server use resumes until the **dead-threshold** is once again reached. **Example: set dead-interval 5** The default setting is **10** seconds.
send-single-label-queries {enabled | disabled}	*Secondary property.* Specifies whether AA-SBC processes queries that have only one label to the configured server. If a query comes in with only one label, AA-SBC appends to the label the static domain (if **use-nnos-domain-in-search** is **enabled**) and any additional domains (if **additional-search-domains** is configured) while processing the query. If this property is **enabled**, the system forward the single-label query to external servers. If it is disabled, the system tries to resolve the query internally but, failing that, does not send it out to an external server. **Example: set send-single-label-queries enabled** The default setting is **disabled**.

server

Purpose

Configures the server(s) to use for DNS and ENUM queries. You can enter any number of servers, and specify, for each, its use (DNS, ENUM, or both). AA-SBC selects which servers will handle a query based on the setting of the resolver **server-scheme** property.

Syntax

```
config vsp dns resolver server ipAddress
```

Properties

Property name	Description
protocol {any | UDP | TCP | TLS}	Sets the protocol the DNS resolver service uses to communicate with the identified server(s). Currently, UDP is the only supported protocol. **Example: set protocol UDP** The default setting is **UDP**.
port *portNumber*	Sets the UDP port number over which the resolver service communicates with the identified server(s). **Example: set port 54** The default setting is **53**.
preference *preference*	Specifies the preference assigned to this server. The lower the value the higher the preference. This value is used if the resolver **server-scheme** property is set to **preference-order**. **Example: set preference 50** The default setting is **100**.

Property name	**Description**
type {dns-only | enum-only | both}	Specifies the type of queries this server will perform. If **both** is not selected, only queries of the specified type are sent to the server. **Example: set type dns-only** The default setting is **both**.
name *string*	Associates a name string with the server configuration. You can then reference this server from other parts of the configuration using this name. For example, the session configuration dns-client-settings objects uses these names to reference servers for client use. **Example: set name corp-server** There is no default setting.

host

Purpose

Statically maps an IP address to a host name. Use this object to more easily manage your DNS configuration by using names instead of addresses. By creating a static configuration for a host name, you prevent a DNS lookup from going out on the wire.

The **host** object requires that you supply a *name* variable. This is the name of an Internet node, for example, a server, a router, or a PC in your network.

Syntax

```
config vsp dns host name
```

Properties

Property name	Description
address *ipAddress*	Sets the IP address to map to the name supplied with the **host** object. **Example: set address 192.168.10.10** There is no default setting.

service

Purpose

Creates a static SIP server-to-service configuration, adding a DNS server resource (SRV) record for each SIP service. (SRV records provide contacts for the specific domain services.) Within each service, you execute this object for each SIP server to establish the order in which to contact them.

The **service** object requires that you supply a *domainName* or *hostName*, a scheme of either SIP or H323, and a protocol. AA-SBC derives the name of the service for which you are configuring server information from these entries.

Syntax

```
config vsp dns service name {sip | h323} {any | UDP | TCP | TLS}
```

Properties

Property name	Description
rule *sipServer* [*port*] [*priority*] [*weight*]	Sets the priority of a SIP server when there are multiple servers configured for a service. This rule sets the criteria for selecting a SIP server for the service derived from the name and protocol entered for the **service** object. Enter: • **SIP-server**—the name of the SIP server for the named service. • **port**—the port on the SIP server through which this service is accessed. • **priority**—the priority of the SIP server. The lower the number, the higher the priority. If two servers have the same priority, the system tries the server with the higher weight first. Enter a value between 1 and 65535. • **weight**—the preference weighting for use when priority settings are equal. The higher the weight, the higher the preference. Enter a value between 0 and 65535. Use a value of 0 when there is only one SIP server configured. **Example: set rule sipServer.companyABC.com 5001 1 10** There is no default setting for the server.The default port is **5060**, the priority is **1**, and the weight is **0**.

naptr

Purpose

Creates a static mapping of service information to a specific host or domain name. AA-SBC uses this information to do a lookup for requests in which it cannot determine either the protocol or port of the destination.

Naming-authority pointer (NAPTR) records are used to set up different services in a domain. They contain rules for converting each request to the correct configured service. Because each transport service over SIP is viewed as a different service (SIP over TCP, UDP, or TLS are each different services), they establish three different NAPTR records. This object configures the preference for use of an appropriate service for each domain. Set one rule for each protocol—UDP, TCP, and TLS. Before a request can be forwarded on, the system must know both the protocol and port for the destination.

The following table describes the decision process for different types of received requests:

The system...		**Resolution**
Knows	**Does not know**	
Protocol Port	None	No lookup is necessary.
Protocol	Port	The system matches on the protocol in the NAPTR records. It then uses that record to identify which service to use. From there, the system does an SRV lookup on the service name to establish the port number.
Port	Protocol or Port Protocol	The system does a NAPTR record lookup based on the port number (if known), starting with the protocol that has the highest priority. If the system cannot find a port match, or does not know the port number, it uses the default protocol (UDP) and the port provided in the original request.

Enter the domain name or host name that you are going to map to a service.

Syntax

```
config vsp dns naptr name
```

Properties

Property name	Description
match {exact | wildcard}	Sets the match criteria for the domain name supplied when you opened the NAPTR object. If set to **exact**, the system only maps to service names that contain an exact match of the domain name you entered. If set to **wildcard**, the system maps to any service name containing the full domain name, but the service name may also contain additional characters to the left of the domain name. For example, if the NAPTR object were opened with the domain name companyABC.com, the service name could match, for example, abc.companyABC.com and xyz.companyABC.com. **Example: set match wildcard** The default setting is **exact**.
rule {any | UDP |TCP | TLS} [*order*] [*preference*]	Sets the lookup procedure for destinations with unknown protocols or ports. You can only have one entry for each protocol within the domain name specified to open the **naptr** object. Enter a rule for each protocol (a total of three rules)—UDP, TCP, and TLS. • **protocol**—the protocol that the rule applies to. Enter either any, UDP, TCP, or TLS. • **order**—the priority of the rule. Use this parameter to set the order in which the system checks The lower the number, the higher priority. If two rules have the same priority, the system uses the rule with the higher weight. Enter a value between 1 and 65535. • **preference**—the preference weighting for use when order settings are equal. The higher the weight, the higher the preference. Enter a value between 0 and 65535. **Example: set rule TLS 5 10** The default protocol is **UDP**, the order (priority) is **10**, and the preference is **50**.

enum-mapping

Purpose

Creates a static configuration mapping between an E.164 number and a host name, providing a static mapping function for unresolvable addresses. The configuration is applied when **enum-operation** property of the dial plan normalization object is **enabled**.

ENUMs are mappings between E.164 (the public network addressing standard) number assignments and URLs. ENUM is a protocol that makes internet resources addressable via a phone number. The protocol uses the DNS cache to identify services available to an E.164 number. By converting E.164 numbers into URLs, AA-SBC uses Enum Server and Naming Authority Pointer (NAPTR) records to look up the services available for a specific E.164 number (via its domain name) in the DNS cache.

The **mapping** object requires that you supply a *phoneNumber* variable. Enter a number for which you want to create a permanent listing in the DNS/ENUM cache. The phone number must be at least four characters long. It is stored in the cache as type NAPTR.

For more information on using DNS to store E.164 numbers, refer to:

- *RFC 3761, The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM)*

Syntax

```
config vsp dns enum-mapping phoneNumber
```

Properties

Property name	Description
domain *name*	Specifies the domain name to append to this phone number mapping. The system will use this domain when the performing an ENUM lookup. By default, the system uses the standard ENUM DNS domain as specified in *RFC 2916, E.164 number and DNS*. **Example: set domain e164.arpa.** The default setting is **e164.arpa.**.
order *priority*	Sets the priority of the mapping when there are multiple entries for a single phone number. The lower the order number, the higher priority. If two mappings have the same priority, the system uses the entry with the higher preference (see below) first. **Example: set order 30** Enter a value between 1 and 65535. The default is **10**.
preference *weight*	Sets the preference weighting for use when order settings are equal. The higher the weight, the higher the preference. **Example: set order 100** Enter a value between 1 and 65535. The default is **50**.

Property name	Description
protocol {UDP | TCP | TLS}	Sets the protocol that should be associated with the telephone number. **Example: set protocol TCP** The default protocol is **UDP**.
replacement *sipURL*	Specifies the name to associate with the phone number. This field is required. You must enter a SIP URL, in the format: `SIP:` *`urlAddress`* The URL address that you enter can be: • a host name only • an IP address only • *`username@hostName`* or *`username@ipAddress`*. See the following SIP specifications for entry format details: • *RFC 3761, The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM)* • *RFC 3966, The tel URI for Telephone Numbers* **Example: set url SIP:jane@company.com** There is no default setting.

reject

Purpose

Statically enters reject entries to the resource records in the cache. These entries are not cleared when the cache otherwise clears, and continually prevent access until the entry is manually removed. Because the record indicates that the entry is a failure, the system does attempt a lookup. (Dynamic reject entries work according to the DNS specification; they are cleared every 5 minutes.)

The **reject** object requires that you supply a *name* and a **type**. Enter a host name, service name, domain name, or IP address. Any request containing that name will be rejected. (See the property descriptions for information on entering wildcards.) Also enter the record type for the entry to be rejected, either:

- **A**—Reject the supplied host name (IPv4 address).
- **PTR**—Reject the supplied IP address (Address-to name-mapping pointer records).
- **SRV**—Reject the supplied service name (Server resource rule).
- **NAPTR**—Reject the supplied domain name (Naming Authority Pointer rule).
- **CNAME**—Reject the canonical name record (makes one domain name an alias of another).
- **NS**—Reject the name server record.

Syntax

```
config vsp dns reject name type
```

Properties

Property name	Description
match {exact | wildcard}	Sets the match criteria for the name supplied when you opened the reject object. If set to **exact**, the system only maps to names that contain an exact match of the name you entered. If set to **wildcard**, the system maps to any name containing the full name, plus any addition characters to the left of the name. Below, if the reject object were opened with the name companyABC.com, the service name could match, for example, tls.companyABC.com and udp.companyABC.com. **Example: set match wildcard** The default setting is **exact**.

dns-server

Purpose

Identifies the IP interface on which the DNS server resides. The DNS server function is defined by the setting of the **mode** property. In **proxy mode**, the DNS server acts as a proxy server, in that it accepts a request, but forwards it to the configured server for fulfillment. In **cache mode**, AA-SBC forwards the request to the resolver. If the resolver has a cached or static entry, it does not forward the request. Instead, it responds to the dns-server with that entry and the server responds to the requestor.

Syntax

```
config cluster box number interface ethX ip name dns-server
config cluster box number interface ethX vlan number ip name dns-server
config box interface ethX ip name dns-server
config box interface ethX vlan number ip name dns-server
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables the DNS server on the current IP interface. **Example: set admin enabled** The default setting is **enabled**.

Property name	Description
port {udp | tcp} [*portNumber*]	Sets the protocol and port used to send and receive DNS requests with the DNS server. **Example: set port tcp 54** The default protocol is **udp**; the known server port number for DNS is port **53**. Enter a port number in the range of 1 to 65535.
mode {cache | proxy *ipAddress* [udp |tcp] [*port*]}	Specifies where the system's DNS server functionality retrieves entries from. If set to **cache**, the default, the system accepts requests, but forwards them to the resolver for fulfillment. (If the resolver has a cached or static entry, it does not forward the request, but responds with the information.) If set to **proxy**, the system forwards the request directly to the specified server. Enter the IP address of the server, as well as the protocol and port for contact. **Example: set mode proxy 10.10.10.10 tcp 222** The default setting is **cache**.

26. DTMF generation objects

DTMF generation description

AA-SBC, if configured to do so with the **auto-conference** property of the session config media object, strips conference codes (DTMF strings) from an INVITE request and injects them into the RTP stream (the established call). This allows AA-SBC to play the DTMF tone digits to a conference server on behalf of the client calling the server. Through this object, you can control the length of play and pause time and volume for the digits that AA-SBC plays.

DTMF generation object summary

The following table lists and briefly describes the **dtmf-generation** object. See the following chapter for other objects in the CLI hierarchy:

- Chapter 75, “Virtual system partition (VSP) objects”

Object name	Description
`dtmf-generation`	Sets playback parameters for the conference codes.

dtmf-generation

Purpose

Sets parameters for the conference codes derived using the **auto-conference** property of the session config media object. AA-SBC applies these parameter settings to the conference codes found in INVITEs, and injects them into the RTP stream.

Syntax

```
config vsp dtmf-generation
```

Properties

Property name	Description
digit-volume *dBm0*	Specifies the volume setting for the conference code tones. The digit volume is measured in decibel (dB) of the measured power referenced to one milliwatt, measured at a zero transmission level point. The smaller the dBm0, the louder the volume. **Example: set digit-volume 25** Enter a value between 1 and 63; the default setting is **20**.
digit-duration *milliseconds*	Specifies the length of time, in milliseconds, that the system plays each digit of the conference code. **Example: set digit-duration** Enter a value between 100 and 10,000; the default setting is **750** milliseconds.
inter-digit-duration *milliseconds*	Specifies the length of time, in milliseconds, that the system pauses between playing each digit in the conference code. **Example: set inter-digit-duration** Enter a value between 0 and 1,000; the default setting is **250** milliseconds.

Property name	Description
pause-duration *milliseconds*	Specifies the number of milliseconds that the system pauses when it encounters a comma character in the conference code. The comma is a special character, written in to the conference code, that indicates the system must wait for the specified time before playing the next tone. **Example: set pause-duration 4500** Enter a value between 500 and 10,000; the default setting is **3000** milliseconds.
as-audio {true | false}	Specifies whether the system sends audio or DTMF packets, when representing conference code tones, to the conference server. When true, the system encodes the sound in the current CODEC (e.g., PCMU or IBC). When false, the system attempts to send DTMF packets. **Example: set as-audio false** The default setting is **true**.

27. Enterprise objects

Enterprise description

Enterprise services work by using an existing directory name service in conjunction with client programs that access that service to look up user entries. By configuring AA-SBC to recognize a particular enterprise service, you are drawing that service under the security protection of AA-SBC, preventing application-level attacks.

Enterprise services are SIP-enabled real-time communication systems and collaboration services. You are configuring AA-SBC so that it can access the required databases to derive the recognized SIP addresses within your enterprise. These services allow an organization to support, among others:

- IP PBX hosted VoIP services
- enterprise instant messaging systems
- mobile devices
- presence-based applications

➡ **Note:** While you can configure enterprise services at any time, you must enable the master-services directory object for AA-SBC to use the service. See Chapter 39, “Master services objects” for more information.

For detailed information AA-SBC enterprise gateways, refer to the *Net-Net OS-E – Session Services Configuration Guide*.

Enterprise object summary

The following table lists and briefly describes the **enterprise** object hierarchy. See the following chapter for other objects in the CLI hierarchy:

- Chapter 75, “Virtual system partition (VSP) objects”

Object name	Description
enterprise	Sets the parameters that allow and define communication with directory service(s) and applications.
directories	Provides access to configuration of the directory services available in your enterprise.
servers	Provides access to configuration of the servers that provide applications in your enterprise.
federations	Provides access to configure junction points between the servers in your enterprise.
unknown-server-policy	Assigns policy to users from unknown SIP addresses.

enterprise

Purpose

Opens the enterprise configuration group object, which is the container for the directory services and application configurations.The enterprise object contains three objects that house these additional configurations—directories, servers, and federations. In addition, it is the parent object of the unknown server policy, which sets the policies to apply to users that are not identified as belonging to a specific server.

Note: You must enable the directory object in master-services before the enterprise object can be become active. See Chapter 39, “Master services objects” for more information.

Syntax

```
config vsp enterprise
```

Properties

Property name	Description
user-group-policy *groupName policyReference*	Specifies the policy to apply to users of any server who are members of the specified group. The group can be either a user group from the directory service schema or a virtual group constructed in the configuration for policy application purposes. Enter a group name, and the system applies the specified policy to any user belonging to that group, regardless of the server for which they are intended. Also enter the complete path to a previously configured policy reference. If this value is set both here and at the server level, through the server object, the system applies both settings. **Example: set user-group-policy lcsAdmin “vsp policies session policies policy noIM”** There is no default setting.

directories

Purpose

Opens the directories configuration object for editing. The directories container includes those objects that represent the directory services available in your enterprise. See the following chapter for a description of each type of enterprise directory and its objects and properties:

- Chapter 23, “Directory objects”

Syntax

```
config vsp enterprise directories
```

Properties

Property name	Description
admin {enabled | disabled}	Specifies whether the directory configurations are available to the system. When **enabled**, the system uses the configuration; when **disabled**, directory services are not available to the system. **Example: set admin enabled** The default setting is **enabled**.

Property name	Description
on-failure {abort | ignore | retry *times sleep* {abort | ignore}}	Sets the system behavior in the event that an enterprise name server is, or becomes, unavailable. Select one of the following: • **abort**—Cancels the current attempt at establishing a session with the communications server and disregards any data generated for other enterprise services. (If one service fails, they all fail.) The system cancels all attempts to bring up any of the configured name services directories. Without a directory loaded, most likely all configured policy will fail. • **ignore**—Ignores the attempt at establishing a session with the failed communications server, but maintains data from other services. • **retry** — Sets the parameters for retrying the name server. Specify: • the number of times, between 1 and 5, to attempt to establish a session with the communications server. The default is **3** attempts. • the interval, in seconds, between attempts. The default is 10 seconds. • the next desired action if all retry attempts fail (either abort or ignore). The default is **ignore**. **Example: set on-failure retry 5 15 abort** The default setting is **ignore**.
resolve-on-update {true | false}	Specifies whether to resolve SIP addresses after a directory update. When set to **true**, the system checks its SIP address database against the updated directory, and changes the address database accordingly. When set to **false**, the system does not change the address database. (You update the directories automatically at boot or by executing the **directory-reset** action.) **Example: set resolve-on-update true** The default setting is **false**.

servers

Purpose

Opens the servers configuration object for editing. The servers container includes those objects that represent the servers that provide applications in your enterprise. See the following chapter for a description of each type of enterprise server and its objects and properties

- Chapter 59, “Server objects”

Syntax

```
config vsp enterprise servers
```

Properties

Property name	Description
default-server *serverReference*	Sets the server to use for all situations in which a packet arrives and does not match any criteria for server selection. Enter the server name as a reference to a previously configured server. **Example: set default-server vsp enterprise servers sametime abcCo** There is no default setting.

federations

Purpose

Opens the federations configuration object for editing. The federations container includes the objects that represent junction points between the servers in your enterprise. A federation is formed by including previously configured servers into the named object. See the following chapter for information on creating a federation:

- Chapter 31, “Federation object”

Syntax

```
config vsp enterprise federations
```

Properties

None

unknown-server-policy

Purpose

Sets the policy to apply to sessions going to or coming from unknown users. You can configure AA-SBC to apply policy when it detects a sender or receiver of a packet that is not registered in the enterprise directory service.

The unknown-server-policy object allows you to specify separate “from” and “to” policies for unregistered users. When configuring this object, you reference previously created policies. See the following for more information on policy:

- Chapter 48, “Policy objects”

Syntax

```
config vsp enterprise unknown-server-policy
```

Properties

Property name	Description
to-policy *policyReference*	Sets the policy to use for all situations in which a packet is destined for a user that is not registered in any enterprise directory service. Enter the policy name as a reference to a previously configured policy. **Example: set to-policy vsp policies session-policies toPolicy** There is no default setting.
from-policy *policyReference*	Sets the policy to use for all situations in which a packet arrives from a user that is not registered in any enterprise directory service. Enter the policy name as a reference to a previously configured policy. **Example: set from-policy vsp policies session-policies fromPolicy** There is no default setting.

28. Eventpush service objects

Eventpush service description

The Eventpush service object enables you to redirect AA-SBC logged events to an external web browser, providing pushlet functionality for use in web services applications. This functionality is only useful in environments where the application uses only Javascript or only PHP. When configured, the feature overcomes a limitation in these languages that prevent them from interacting with a Java-based pushlet. (For web services applications written in Java, configuration of this object is necessary.)

The pushlet included in AA-SBC software allows users of a web services application to have the events applicable to their call returned to their web browser. When a sample application receives an event via WSDL, it passes the event to the pushlet. Because the pushlet sorts events by ID, the web browser can request events by pushlet ID to maintain current call status.

Eventpush object summary

The following table lists and briefly describes the **eventpush-service** object. See the following chapter for other objects in the CLI hierarchy:

- Chapter 14, “Cluster, box, and interface objects”
- Chapter 77, “VLAN objects”
- Chapter 35, “IP objects”

Object name	Description
`eventpush-service`	Configures and redirects AA-SBC logged events to an external Web browser.

eventpush-service

Purpose

Configures and redirects AA-SBC logged events to an external web browser over an HTTP or HTTPS web service port. It enables events to be asynchronously sent to clients. The eventpush-service requires the external-services event-group to declare the destination service URL of the external device and the preferences pushlet-app property to configure the page domain.

Perform the following steps to enable and configure the **eventpush-service**.

1. Configure the **preferences** pushlet-app object and set the **domain-page** property to the common domain name of the target system running the web application.

```
CXC> config preferences
config preferences> config pushlet-app
config pushlet-app> set page-domain companyABC.com
```

2. Configure the IP **eventpush-service**. This creates the process in AA-SBC that responds top client event requests.

```
CXC> config box
config box> config interface eth3
config interface eth3> config ip eventpush
Creating ‘ip eventpush’
config ip eventpush> config eventpush-service
config eventpush-service> set admin enabled
config eventpush-service> set protocol http 8081
```

3. Edit the target web application to include an IFrame. The IFrame is comprised of the name of the AA-SBC device running the eventpush-service application, the eventpush service port, and the string **/cometapp/covergence.html**.

   For example, if the name of the AA-SBC device running the eventpush-service is *xyz.com* with the eventpush web service running on port 8081, and if the system is running over HTTP, then the reference is **http://xyz.com:8081/cometapp/covergence.html**.

4. Configure the external-services\event-group event-service **service-url** property, so that events are passed to the destination eventpush web service. Enter the destination domain IP, the eventpush service port, and the context string /**cometapp/callouts**. For example, **http://127.0.0.0:8081/cometapp/callouts**.

```
CXC> config external-services
```

```
config external-services> config event-group a
config event-group a> config event-service cometd
Creating ‘event-service cometd’
config event-service cometd> set service-url http://127.0.0.0:8081/
cometapp/callouts
```

Syntax

```
config cluster box number interface ethX ip name eventpush-service
config cluster box number interface ethX vlan number ip name
   eventpush-service
config box interface ethX ip name eventpush-service
config box interface ethX vlan number ip name eventpush-service
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables the eventpush service. Enabling the service allows you to redirect AA-SBC logged events to an external Web browser. **Example: set admin disabled** The default setting is **enabled.**
protocol {http *port* | https *port* [*certificateReference*] [*alias*] }	Sets the protocol to use for pushlet operations. After setting a protocol, you can select the pushlet listen port (or accept the default). This is the port over which the server listens for HTTP(S) requests. Select either: • **http**—sets an insecure (unencrypted) protocol for use in web transmission. • **https**—provides secure transmission of web pages by using HTTP over SSL. Optionally, you can set: • a reference to a previously configured certificate (configured with the certificate object). • an alias for the key in the key store (named in the certificate configuration). **Example: set protocol https 8444”vsp tls certificated cxc.company.com” certKey** The default protocol is **http** with a port setting of 8081. If you set the protocol to **https,** the default port 8443.
max-threads *integer*	Specifies the maximum number of total workder threads, both active and spare (idle), allocated to the web server (eventpush service). **Example: set max-threads 15** Enter a value between 1 and 500; the default setting is **10** threads**.**

Property name	**Description**
min-spare-threads *integer*	Specifies the minimum number of inactive threads that the system must leave allocated to the web server. When the system removes idle threads, it must leave this number of spares available. **Example: set min-spare-threads 3** Enter a value between 0 and 50; the default setting is **1** thread**.**
max-spare-threads *integer*	Specifies the maximum number of inactive threads the system can leave allocated to the eventpush service. When the system detects idle threads, it cannot have more than this number. **Example: set max-spare-threads 8** Enter a value between 0 and 50; the default setting is **5** threads.
page-domain	Specifies the common domain name of the AA-SBC and the system running the web application. **Example: set page-domain www.acme.com**

Eventpush service objects

29. External services objects

External services description

The external-services object defines various external web services that provide, or act as receptacle for, information to and from AA-SBC. A web service provides interoperability between platforms and operating systems. Because the public interface is described in the Web Services Description Language (WSDL), an XML-based service description language, disparate platforms can use the web services to exchange data.

You can configure AA-SBC as both a WSDL client and server. Use the external-services object to configure it as a client; use the web-service object to enable the interface, allowing AA-SBC to function as a server.

In AA-SBC, external web services can be used to share location databases, apply session policy, and track system notifications. See *Net-Net OS-E – Using the NNOS-E Management Tools* for more information on WSDL.

External services object summary

The following table lists and briefly describes the **external-services** object:

Object name	Description
`external-services`	Provides access to configuring remote web services.
`policy-group`	Creates a group to which the individual external policy service servers belong.
`policy-service`	Specifies the URL of a server maintaining session policy configurations.
`location-group`	Creates a group to which the individual external location service servers belong.
`red-sky-location-service`	Specifies the URL of the RedSky location information web services.

Object name	Description
tcs-location-service	Specifies the URL of the TCS location information web services.
generic-service	Specifies the URL of the non-RedSky or TCS location information web services.
event-group	Creates a group to which the individual external event service servers belong.
event-service	Specifies the URL of a server tracking system notifications.

external-services

Purpose

Opens the external-services object. Within the subobjects, you configure the URLs used to access the remote web services. Configuring external services configures AA-SBC as a client, which allows it to make calls out to other web service endpoints that implement the call out interfaces. The call out interfaces can be used to provide location information and/or policy information.

Syntax

```
config external-services
```

Properties

Property name	Description
policy-services-type {old | new}	Specifies the method by which AA-SBC sends request and processes responses for WSDL policy call-outs. Select one of the following options: • old—Select this option to use the AA-SBC's older Java-code method. This method does not support persistent TCP connections and has to connect and disconnect for each request. • new—Select this option to use the AA-SBC's newer C-code method. This method is faster and more reliable than the old method and is able to handle persistent connections, sending multiple requests on a single TCP connection. **Example: set policy-services-type new** The default setting is **old**.

policy-group

Purpose

Creates a group to which the individual external policy service servers belong. AA-SBC applies the configured failover mechanism to all server configurations within this group. When a request is made, AA-SBC searches the list of servers in the group sequentially for one that is available. If it does not find an available server, it does not send out the request.

When AA-SBC detects that a server is unavailable, it changes that web service server status to unavailable. (You can verify the status of a server through the **show web-services-callout-details** command with the **availability** field.) If the status is unavailable, you must set it to available (once it is) with the web-services action before AA-SBC can use that server again.

Syntax

```
config external-services policy-group name
```

Properties

Property name	Description
failover-detection {none | passive [*response*] [*failures*] | active [*interval*] [*failureInterval*] [*retries*] [*response*] [*failures*]}	Specifies the type of failure detection the system should perform for all external policy servers in this group. Select either: • **none**—the system does no failover detection. If a request is not serviced, the system continues to send requests until a configured timeout value is reached or the request is manually withdrawn. • **passive**—sets a number of failures or a timeout period before that the system can experience before it determines that the server is unavailable. The passive method only detects failures in response to a service request. Set the following: • *response*—the number of milliseconds that the system waits for a response from the server. When the timer expires, the system classifies the server as unavailable. Enter a value from 10 to 60,000. The default is 60,000 seconds. • *failures*—the number of failures that the system allows before determining that the server is unavailable. Enter a value from 1 to 100,000. The default is 3 failures. *continued*

Property name	Description
failure-detection *continued*	- **active**—uses the heartbeat URL assigned with the policy-, location- or event-service objects to constantly test server availability. Additional parameters to this property allow detection in the event that the heartbeat location is available but the web service is not. Set the following: - *interval*—the number of seconds between requests sent to the **heartbeat-url**. Enter a value 1 to 600. The default is 10 seconds. - *failureInterval*—the number of seconds to wait between request attempts to the **heartbeat-url** when the previous request failed. Enter a value 1 to 10. The default is 1 seconds. - *retries*—the number of unsuccessful requests to the **heartbeat-url** the system allows before determining that the server is unavailable. Enter a value 0 to 10. The default is 1 retry. - *response* and *failures*—see the **passive** description. **Example: set failover-detection active 20 2 2** The default setting is **none**.
max-queue-length *requests*	Sets the maximum number of WSDL requests that can be queued for a policy group (awaiting assignment to a server). If the queue grows to this number, subsequent requests are rejected, with the result “queue-clipped,” until the queue drops below this level. **Example: set max-queue-length 32** The default setting is **64** requests.

Property name	Description
connection-mode {persistent *seconds page* | lingering | transient}	Specifies the manner in which connections between AA-SBC and the WSDL server are established and maintained. Select either: • **persistent**—connections are initiated at boot time, and maintained using periodic keepalives. Specify an inactivity timeout, between 2 and 120 seconds, and a keepalive page. • **lingering**—connections are made on demand, then linger until broken by the remote server. • **transient**—connections are made on demand, then broken when a response is received. **Example: set connection-mode lingering** The default setting is **persistent**; the default inactivity time is 10 seconds and the default page is /covws/callouts?wsdl.
overall-request-timeout *seconds*	Specifies the number of seconds a request can remain in the queue for a policy server before it is timed out by AA-SBC. **Example: set overall-request-timeout 10** Enter a value from 1 to 30; the default setting is **5** seconds.
connection-count *seconds*	Specifies the number of seconds a request can remain in the queue for a policy server before it is timed out by AA-SBC. **Example: set overall-request-timeout 10** The default setting is **5** seconds.

policy-service

Purpose

Sets the URL of the external server that maintains policy configurations to apply to a session. When configured, AA-SBC calls out to the specified server with session data, and the server returns the appropriate policy configuration, which AA-SBC then applies to the session. To open this object, enter a name for the policy server configuration.

Syntax

```
config external-services policy-group name policy-service name
```

Properties

Property name	Description
admin {enabled | disabled}	Specifies whether this policy service server is enabled for use. **Example: set admin disabled** The default setting is **enabled**.
service-url *url*	Specifies the URL of the remote web service that maintains the session configuration. **Example: set service-url http://myserver:8080/ myPolicyServer** There is no default setting.
heartbeat-url *url*	Specifies a location on the external server that can be used to test server connectivity. Typically, you would specify a pointer to static content on the server (e.g., a small file). The system then requests the file from the external service to determine server availability. This property is used when the **failover-detection** property is set to **active**. **Example: set heartbeat-url http://myserver:8080/ heartbeatTest.html** There is no default setting.

Property name	Description
connect-timeout *milliseconds*	Specifies the length of time, in milliseconds, that the system allows to complete a connection to the external policy service before cancelling the request. **Example: set connect-timeout 1000** Enter a value between 100 and 30,000; the default setting is **500** milliseconds.
read-timeout *milliseconds*	Specifies the length of time, in milliseconds, that the system waits for a response from the external policy service before cancelling the request. **Example: set read-timeout 1500** Enter a value between 100 and 30,000; the default setting is **2000** milliseconds.
priority *priority*	Specifies the priority of this server within the policy group. The lower the number, the higher the priority. **Example: set priority 5** Enter a value between 1 and 99; the default setting is **1**.
connection-count *count*	Specifies the number of simultaneous connections allowed to this server. Multiple connections can improve performance. **Example: set connection-count 2** Enter a value between 1 and 16; the default setting is **1**.

location-group

Purpose

Creates a group to which the individual external location service servers belong. AA-SBC applies the configured failover mechanism to all server configurations within this group. When a request is made, AA-SBC searches the list of servers in the group sequentially for one that is available. If it does not find an available server, it does not send out the request.

When AA-SBC detects that a server is unavailable, it changes that web service server status to unavailable. (You can verify the status of a server through the **show web-services-callout-details** command with the **availability** field.) If the status is unavailable, you must set it to available (once it is) with the web-services action before AA-SBC can use that server again.

Syntax

```
config external-services location-group name
```

Properties

Property name	Description
failover-detection {none | passive [*response*] [*failures*] | active [*interval*] [*failureInterval*] [*retries*] [*response*] [*failures*]}	Specifies the type of failure detection the system should perform for all external location servers in this group. Select either: • **none**—the system does no failover detection. If a request is not serviced, the system continues to send requests until a configured timeout value is reached or the request is manually withdrawn. • **passive**—sets a number of failures or a timeout period before that the system can experience before it determines that the server is unavailable. The passive method only detects failures in response to a service request. Set the following: • *response*—the number of milliseconds that the system waits for a response from the server. When the timer expires, the system classifies the server as unavailable. Enter a value from 10 to 60,000. The default is 60,000 seconds. • *failures*—the number of failures that the system allows before determining that the server is unavailable. Enter a value from 1 to 100,000. The default is 3 failures. *continued*

Property name	Description
`failure-detection` *continued*	• **active**—uses the heartbeat URL assigned with the policy-, location- or event-service objects to constantly test server availability. Additional parameters to this property allow detection in the event that the heartbeat location is available but the web service is not. Set the following: • *interval*—the number of seconds between requests sent to the **heartbeat-url**. Enter a value 1 to 600. The default is 10 seconds. • *failureInterval*—the number of seconds to wait between request attempts to the **heartbeat-url** when the previous request failed. Enter a value 1 to 10. The default is 1 seconds. • *retries*—the number of unsuccessful requests to the **heartbeat-url** the system allows before determining that the server is unavailable. Enter a value 0 to 10. The default is 1 retry. • *response* and *failures*—see the **passive** description. **Example: set failover-detection passive 45000 3** The default setting is **none**.

red-sky-location-service

Purpose

Configures the URL of the VoIP Positioning Center (VPC) providing location services (caller location) for VoIP subscribers using Location Information Services (LIS) from RedSky Technologies, Inc. If the VPC returns a location record from the WSDL query, or a message that indicates that a location record exists, the call registration completes and AA-SBC forwards the call. Otherwise, the call is declined.

Syntax

```
config external-services location-group name red-sky-location-service
   name
```

Properties

Property name	Description
admin {enabled | disabled}	Specifies whether this location service server is enabled for use. **Example: set admin disabled** The default setting is **enabled**.
service-url *url*	Specifies the URL of the location web services provided by RedSky Technologies. **Example: set service-url http://e911_RedSky@providerOne.com** There is no default setting.
heartbeat-url *url*	Specifies a location on the external server that can be used to test server connectivity. Typically, you would specify a pointer to static content on the server (e.g., a small file). The system then requests the file from the external service to determine server availability. This property is used when the **failover-detection** property is set to **active**. **Example: set heartbeat-url http://e911_RedSky@providerOne.com/heartbeatTest.html** There is no default setting.

Property name	Description
connect-timeout *milliseconds*	Specifies the length of time, in milliseconds, that the system allows to complete a connection to the external location service before canceling the request. If the connection times out, the user record is still added to the location cache, but as an unregistered user. **Example: set connect-timeout 1000** Enter a value between 100 and 30,000; the default setting is **500** milliseconds.
read-timeout *milliseconds*	Specifies the length of time, in milliseconds, that the system waits for a response from the external location service before canceling the request. If the read timer expires, the user record remains unverified in the location cache. **Example: set read-timeout 1500** Enter a value between 100 and 30,000; the default setting is **2000** milliseconds.

tcs-location-service

Purpose

Configures the URL of the VoIP Positioning Center (VPC) providing location services (caller location) for VoIP subscribers using location-based E911 services provided by TeleCommunications Systems, Inc. If the VPC returns a location record from the WSDL query, or a message that indicates that a location record exists, the call registration completes and AA-SBC forwards the call. Otherwise, the call is declined.

Syntax

```
config external-services location-group name tcs-location-service name
```

Properties

Property name	Description
admin {enabled | disabled}	Specifies whether this location service server is enabled for use. **Example: set admin disabled** The default setting is **enabled**.
service-url *url*	Specifies the URL of the location web services provided by TeleCommunications Systems. **Example: set service-url http://e911_TCS@locationServer.com** There is no default setting.
heartbeat-url *url*	Specifies a location on the external server that can be used to test server connectivity. Typically, you would specify a pointer to static content on the server (e.g., a small file). The system then requests the file from the external service to determine server availability. This property is used when the **failover-detection** property is set to **active**. **Example: set heartbeat-url http://e911_TCS@locationServer.com/heartbeatTest.html** There is no default setting.

Property name	Description
connect-timeout *milliseconds*	Specifies the length of time, in milliseconds, that the system allows to complete a connection to the external location service before canceling the request. If the connection times out, the user record is still added to the location cache, but as an unregistered user. **Example: set connect-timeout 1000** Enter a value between 100 and 30,000; the default setting is **500** milliseconds.
read-timeout *milliseconds*	Specifies the length of time, in milliseconds, that the system waits for a response from the external location service before canceling the request. If the read timer expires, the user record remains unverified in the location cache. **Example: set read-timeout 1500** Enter a value between 100 and 30,000; the default setting is **2000** milliseconds.

generic-service

Purpose

Configures the URL of the VoIP Positioning Center providing location services for VoIP subscribers using services other than RedSky Technologies or TeleCommunications Systems. If the VPC returns a location record from the WSDL query, or a message that indicates that a location record exists, the call registration completes and AA-SBC forwards the call. Otherwise, the call is declined.

Syntax

```
config external-services location-group name generic-service name
```

Properties

Property name	Description
admin {enabled | disabled}	Specifies whether this location service server is enabled for use. **Example: set admin disabled** The default setting is **enabled**.
service-url *url*	Specifies the URL of the remote web services providing location information. **Example: set service-url http://loc_rcrds@otherLocations.com** There is no default setting.
heartbeat-url *url*	Specifies a location on the external server that can be used to test server connectivity. Typically, you would specify a pointer to static content on the server (e.g., a small file). The system then requests the file from the external service to determine server availability. This property is used when the **failover-detection** property is set to **active**. **Example: set heartbeat-url http://loc_rcrds@otherLocations.com/heartbeatTest.html** There is no default setting.

Property name	Description
connect-timeout *milliseconds*	Specifies the length of time, in milliseconds, that the system allows to complete a connection to the external location service before canceling the request. If the connection times out, the user record is still added to the location cache, but as an unregistered user. **Example: set connect-timeout 1000** Enter a value between 100 and 30,000; the default setting is **500** milliseconds.
read-timeout *milliseconds*	Specifies the length of time, in milliseconds, that the system waits for a response from the external location service before canceling the request. If the read timer expires, the user record remains unverified in the location cache. **Example: set read-timeout 1500** Enter a value between 100 and 30,000; the default setting is **2000** milliseconds.

event-group

Purpose

Creates a group to which the individual external event service servers belong. AA-SBC applies the configured failover mechanism to all server configurations within this group. When a request is made, AA-SBC searches the list of servers in the group sequentially for one is that is available. If it does not find an available server, it does not send out the request.

When AA-SBC detects that a server is unavailable, it changes that web service server status to unavailable. (You can verify the status of a server through the **show web-services-callout-details** command with the **availability** field.) If the status is unavailable, you must set it to available (once it is) with the web-services action before AA-SBC can use that server again. When a request is made, AA-SBC searches the list of servers in the group sequentially for one that is available. If it does not find an available server, it does not send out the request.

Syntax

```
config external-services event-group name
```

Properties

Property name	Description
failover-detection {none | passive [*response*] [*failures*] | active [*interval*] [*failureInterval*] [*retries*] [*response*] [*failures*]}	Specifies the type of failure detection the system should perform for all external event servers in this group. Select either: • **none**—the system does no failover detection. If a request is not serviced, the system continues to send requests until a configured timeout value is reached or the request is manually withdrawn. • **passive**—sets a number of failures or a timeout period before that the system can experience before it determines that the server is unavailable. The passive method only detects failures in response to a service request. Set the following: • *response*—the number of milliseconds that the system waits for a response from the server. When the timer expires, the system classifies the server as unavailable. Enter a value from 10 to 60,000. The default is 60,000 seconds. • *failures*—the number of failures that the system allows before determining that the server is unavailable. Enter a value from 1 to 100,000. The default is 3 failures. *continued*

Property name	Description
failure-detection *continued*	• **active**—uses the heartbeat URL assigned with the policy-, location- or event-service objects to constantly test server availability. Additional parameters to this property allow detection in the event that the heartbeat location is available but the web service is not. Set the following: • *interval*—the number of seconds between requests sent to the **heartbeat-url**. Enter a value 1 to 600. The default is 10 seconds. • *failureInterval*—the number of seconds to wait between request attempts to the **heartbeat-url** when the previous request failed. Enter a value 1 to 10. The default is 1 seconds. • *retries*—the number of unsuccessful requests to the **heartbeat-url** the system allows before determining that the server is unavailable. Enter a value 0 to 10. The default is 1 retry. • *response* and *failures*—see the **passive** description. **Example: set failover-detection active 10 1 1 45000 3** The default setting is **none**.
trap-filter {generic | csta | dos | sip | system | tls}	Specifies which categories of SNMP traps the system sends out the WSDL interface to the external event service server. You can set as many of the pre-configured trap categories as necessary. If you do not set any trap filters, the system sends all traps. Use the **show trap-categories** command to list the possible trap types in each category. **Example: set trap-filter sip** There is no default setting.

event-service

Purpose

Configures the URL of a server used for tracking AA-SBC events. (These events are similar to SNMP traps.) To open this object, enter a name for the server configuration.

Syntax

```
config external-services event-group name event-service name
```

Properties

Property name	Description
admin {enabled | disabled}	Specifies whether this event service server is enabled for use. **Example: set admin disabled** The default setting is **enabled**.
service-url *url*	Specifies the URL of the remote server tracking system notifications. **Example: set service-url http://myEventServer:8080/myNotifications** There is no default setting.
heartbeat-url *url*	Specifies a location on the external server that can be used to test server connectivity. Typically, you would specify a pointer to static content on the server (e.g., a small file). The system then requests the file from the external service to determine server availability. This property is used when the **failover-detection** property is set to **active**. **Example: set heartbeat-url http://myEventServer:8080/heartbeatTest.html** There is no default setting.

Property name	Description
connect-timeout *milliseconds*	Specifies the length of time, in milliseconds, that the system allows to complete a connection to the external event service before canceling the request. **Example: set connect-timeout 1000** Enter a value between 100 and 30,000; the default setting is **500** milliseconds.
read-timeout *milliseconds*	Specifies the length of time, in milliseconds, that the system waits for a response from the external event service before canceling the request. **Example: set read-timeout 1500** Enter a value between 100 and 30,000; the default setting is **2000** milliseconds.

30. Features licensing objects

Features description

The AA-SBC system and management software uses a license to provide you with the specific features and capacity you requested. When you purchased AA-SBC, you selected the features or bundles of features desired. You were then provided with a license file that enabled those features and set the permissible number of sessions or endpoints.

The **features** object allows you to modify the session/endpoint values. Note that you can not exceed the value specified in the license. To obtain more capacity, contact Technical Support or your sales representative.

Features object summary

The following table lists and briefly describes the **features** object.

Object name	Description
features	Modifies capacity settings for licensed software features.

features

Purpose

Temporarily modifies your existing AA-SBC software license to support a different capacity (either number of sessions or endpoints). You can only set the capacity for a session/endpoint to a value lower than the number allowed by your license. For example, you may have a license that allows 5000 IM sessions. You can use this configuration object to temporarily allow only 3000 sessions. You can also reset the feature to full capacity through this object.

Note that the properties you see at the CLI depend on the software features that your license supports. There are additional “non-royalty” CODECs that the system supports but that are also not displayed in the CLI. (These field values appear greyed-out when displaying the feature list from the AA-SBC Management System). You must obtain a license update to make those feature values configurable. The **Properties** table provides a complete list of licensable software features in AA-SBC.

The default value for all properties is the maximum capacity permitted by your license.

Syntax

```
config features
```

Properties

Property name	Description
signaling-sessions *integer*	Enables the system to perform stateful, application-level inspection, processing, and routing of SIP signaling streams. To be operational, a AA-SBC cluster requires a signaling and/or media processing software license.
media-sessions *integer*	Enables the system to anchor and route SIP-associated media streams (audio, video, file transfer, etc.). To be operational, a AA-SBC cluster requires a signaling and/or media processing software license. Also, each system media proxy must be controlled by a system signaling proxy, which may be co-resident in the same AA-SBC chassis as the signaling proxy or in a different system chassis.

Property name	Description
instant-message-and-presence-sessions *integer*	Enables the system to perform stateful, application-level inspection, processing, and routing of SIP/SIMPLE-based instant messaging and presence traffic. The IM and presence proxy must be co-resident with a signaling proxy on the same system chassis.
high-availability-sessions *integer*	Enables the system to participate in an active-active or active-standby high-availability cluster.
authentication-access-sessions *integer*	Validates the identities of users and/or domains cryptographically, preventing unauthorized users from gaining access to network resources. Integrates with existing authentication and credentialing systems via standard protocols (RADIUS, PKI, DIAMETER, etc.).
signaling-encryption-sessions *integer*	Encrypts and decrypts SIP signaling message headers and bodies using TLS. This ensures the authenticity, confidentiality, and integrity of SIP signaling streams.
media-encryption-sessions *integer*	Encrypts and decrypts SIP-associated media sessions (audio, video, file transfer, etc.) using the Secure Real-time Transport Protocol (SRTP). This ensures the authenticity, confidentiality, and integrity of real-time media information.
media-validation-sessions *integer*	Ensures that the media sessions set up by a SIP user agent is the same as the session that was negotiated during the associated signaling dialogs and permitted by media control policies. This prevents both attacks that exploit the independence of SIP signaling and media channels and unauthorized consumption of bandwidth.
dos-protection-sessions *integer*	Detects and mitigates brute force (resource exhaustion) DOS attacks. Monitors short-term network, transport, and application-level connection activity, detects abnormal aggregate signaling patterns, and denies resources to sessions that match the attack profile.
session-admission-control-sessions *integer*	Limits calling activity based on administratively defined thresholds for session count, total bandwidth, and/or observed quality of service (QOS) metrics. Session admission control policies can be defined on logical and/or physical network interfaces.

Property name	**Description**
media-control-sessions *integer*	Enables fine-grained control over SIP-associated media sessions. For example, an administrator could define a policy saying that only people in a particular group or department (as defined in the directory) can do video sessions.
qos-control-sessions *integer*	Enables control of the QOS of SIP-based applications by performing policy based L2/L3 QOS marking.
session-routing-control-sessions *integer*	Implements intelligent session routing policies such as application-aware load balancing and inbound call routing (e.g., parallel fork, sequential search, presence based routing, and others).
file-transfer-control-sessions *integer*	Enables policy-based control over SIP file transfers.
instant-message-content-control-sessions *integer*	Enforces acceptable use policies on the content of instant messages. Scans IM content for string patterns matching regular expressions and takes a policy-defined action when it finds a match.
url-control-sessions *integer*	Controls propagation of URLs embedded in SIP-based instant messages. This enables enforcement of acceptable use policies and prevents the propagation of blended threats via SIP applications.
session-detail-recording-sessions *integer*	Creates detailed records for signaling sessions that traverse AA-SBC. From this you can track the usage of SIP-based services and applications.
qos-detail-recording-sessions *integer*	Creates QOS records for media sessions that traverse the system, providing data for network engineering, capacity planning, and troubleshooting
audio-recording-sessions *integer*	Enables policy-based recording of audio session content, demonstrating compliance with electronic communications monitoring policy and regulation.
video-recording-sessions *integer*	Enables policy-based recording of video session content, demonstrating compliance with electronic communications monitoring policy and regulation.
file-recording-sessions *integer*	Enables policy-based recording of file transfer session content, demonstrating compliance with electronic communications monitoring policy and regulation.

Features licensing objects

Property name	Description
file-mirror-db-size *integer*	Enables file mirroring and sets the number of files the system can concurrently mirror.
instant-message-recording-sessions *integer*	Enables policy-based recording of IM session content, demonstrating compliance with electronic communications monitoring policy and regulation.
nat-traversal-sessions *integer*	Enables SIP-based applications to traverse remote NAT/firewall functions that may not be under the organization's authority or control. This extends SIP-based applications and services to remote endpoints.
directory-integration-sessions *integer*	Enables the system to import information from directories with Lightweight Directory Access Protocol (LDAP) interfaces so that administrators can define and enforce directory-based policies.
lcs-sametime-gateway-sessions *integer*	Enables presence visibility and messaging connectivity between Microsoft Live Communication Server (LCS) and IBM Lotus Sametime communities.
transcode-sessions *integer*	Enables transcoding media types, which is the process of converting media from one CODEC into a different CODEC on output.
provisional-transcode-sessions *integer*	Enables provisional transcoding sessions. These sessions are temporary sessions used until the system has established whether a call needs transcoding. If it does require transcoding, the session then uses the **transcode-sessions** license.

Property name	Description
g723 *integer* g728 *integer* g729 *integer* g726-16 *integer* g726-24 *integer* g726-32 *integer* g726-40 *integer* gsm integer gsm-amr *integer* ilbc *integer* g722-1 *integer*	Sets the maximum number of concurrent CODEC sessions available for encoding/decoding. The limit does not apply to forwarding or recording RTP packets, only to: • making a transcoded call (see Transcoding media types for more information) • playing recorded announcements • mixing recorded calls for playback • playing back of recorded calls • archiving calls. Note that if you have the media object **music-on-hold** property enabled, AA-SBC holds open one license seat for playing the music until the rtp-cache is released (e.g., system reboot or rtp-header action). This applies to each CODEC type used for playing music on hold. For other announcements and DTMF generation (as audio), the system requires one license until the entire announcement or tone is cached, and the seat is released. The default setting for these properties is the maximum number of allowable licenses. For g723, g729, gsm-amr, and g722-1, the number of available license seats is more restrictive. Contact Technical Support for additional capacity.
monitoring-calls *integer*	Sets the maximum number of monitored calls allowed concurrently.
csta-sessions *integer*	Sets the maximum number of concurrent CSTA sessions allowed.

31. Federation object

Federation description

Federations are groupings of server configurations that allow users of that server group to communicate. They provide the ability to connect across organizations. For example, one LCS deployment could communicate with another, or an LCS deployment could communicate with a Sametime setup. The two organizations, by joining the federation, designate each other as trusted federated partners. The network configuration requires each partner in a federation to communicate through a connecting device, which routes all SIP traffic across the network. (All external traffic passes through the device before entering the internal network.)

By creating federations in AA-SBC, you allow users in different domains to communicate. AA-SBC acts as the device in the middle (an Access Proxy for LCS, a SIP connector for Sametime, for example).

➡ **Note:** You must set the **admin** property of the **precall-authorication** object to **enabled** if you are implementing a Sametime-to-LCS federation. If you are running Sametime-to-Sametime or LCS-to-LCS, the **admin** property must be **disabled**.

You add servers that have been previously configured. See the following chapters for information on the specific server types:

- Chapter 59, “Server objects”

Federation object summary

The following table lists and briefly describes the **federation** object. See the following chapters for other objects in the CLI hierarchy:

- Chapter 75, “Virtual system partition (VSP) objects”
- Chapter 27, “Enterprise objects”

Object name	Description
`federation`	Opens the federation configuration object to allow you to add configured servers to the federation.

`federation`

Purpose

Opens or creates a federation instance for configuration. Within this federation, you add the servers that view each other as trusted partners.

Syntax

```
config vsp enterprise federations federation name
```

Properties

Property name	Description
server *`serverReference`*	Specifies the name of the server configurations that you want included in the federation. You must specify the complete path name for a previously configured server. **Example: set server vsp enterprise server lcs lcs-server** There is no default setting.

Property name	Description
default-policy *policyReference*	Sets the name of the policy to apply if no more specific policy is in place. Enter a previously configured policy reference. **Example: set default-policy vsp policies session-policies policy lcs** There is no default setting.
user-group-policy *groupName* *policyReference*	Specifies the policy to apply to users of this server who are members of the specified group. The group can be either a user group from the directory service schema or a virtual group constructed in the configuration for policy application purposes. Enter a group name, and the system applies the specified policy to any user belonging to that group. Also enter the complete path to a previously configured policy reference. If this value is set both here and at the enterprise level, through the enterprise object, the system applies both settings. **Example: set user-group-policy lcsAdmin "vsp policies session policies policy noIM"** There is no default setting.

Federation object

32. H.323 objects

H.323 description

H.323 is a widely-deployed multimedia conferencing protocol which includes voice, video, and data conferencing for use over packet switched networks. AA-SBC acts as a peer Gatekeeper on a H.323 system, supporting Gatekeeper-Routed Signaling or direct endpoint signaling. This object enables H.323 on an interface and sets the listening ports. Use the server object to define the H.323 gateway.

H.323 object summary

The following table lists and briefly describes the **h323** objects. See the following chapters for other objects in the CLI hierarchy:

- Chapter 14, “Cluster, box, and interface objects”
- Chapter 77, “VLAN objects”
- Chapter 35, “IP objects”

Object name	Description
h323	Configures the H.323 port settings.

h323

Purpose

Sets the port(s) on the system to listen for H.323 connections and RAS messages, and enables the protocol.

Syntax

```
config cluster box number interface ethX ip name h323
config cluster box number interface ethX vlan number ip name h323
config box interface ethX ip name h323
config box interface ethX vlan number ip name h323
```

Properties

Property name	Description
admin {enabled | disabled}	Sets the administrative state of the H.323 protocol, either **enabled** (running) or **disabled**. Use this property to enable or disable the protocol on the interface. Use the **port** property to set the state for an individual listener. **Example: set admin enabled** The default setting is **enabled**.
port {enabled | disabled} *port*	Sets the port over which the appliance listens for H.323 messages. Because you can set multiple ports to listen for H.323 connections, you can also use this property to, for example, disable specific ports without disabling the protocol. **Example: set port 1721** Enter a port number between 1 and 65535. The default port is **1720**.
GKport {enabled | disabled} *port* {any | UDP | TCP | TLS} *identifier*	Sets the port over which the appliance listens for RAS (H.225 Registration, Admission, and Status) messages. Set an administrative status (**enabled** by default), a port number, a transport protocol (**UDP** by default) and unique string to identify the listening port. The protocol you select should be the one used for transporting RAS messages (UDP in accordance with the specification). **Example: set GKport enabled 1720 UDP gk1** The default port is **1719**.

33. ICMP object

ICMP description

The *Internet Control Message Protocol (ICMP)*, defined in RFC 792, is a protocol used to determine whether a destination is unreachable. A TCP/IP-based protocol, ICMP verifies, through error and control messages between a host and an Internet gateway, the validity of an IP address. For example, ICMP functions are used by ping utilities to verify network connectivity. You configure ICMP for each IP interface that requires the functionality.

ICMP object summary

The following table lists and briefly describes the **icmp** object. See the following chapters for other objects in the CLI hierarchy:

- Chapter 14, “Cluster, box, and interface objects”
- Chapter 77, “VLAN objects”
- Chapter 35, “IP objects”

Object name	Description
`icmp`	Configures ICMP communications settings.

icmp

Purpose

Configures the ICMP protocol on the IP interface that hosts it.

Syntax

On a public IP interface:

```
config cluster box number interface ethX ip name icmp
config cluster box number interface ethX vlan number ip name icmp
config box interface ethX ip name icmp
config box interface ethX vlan number ip name icmp
```

Properties

Property name	Description
admin {enabled | disabled}	Sets the administrative state of the ICMP protocol, either **enabled** (running) or **disabled**. When disabled, you can still configure the ICMP parameters, but the parameters do not become active until the **admin** property is set to **enabled**. **Example: `set admin enabled`** The default setting is **enabled**.
limit	Limits the number of ICMP packets that can be received per second on this IP interface. **Example: set limit 50** Min: 1 / Max: 1000 The default setting is **10**.

34. IM Filtering objects

IM filtering description

AA-SBC can scan all SIP Instant Message (IM) content, comparing IM message text (or a list of URLs extracted from the message) with:

- a list of words or regular expressions
- a list of URLs

Use the IM filtering object to create the list of words, regular expressions, or URLs to match against. When a match occurs to one of your lists, AA-SBC can take a variety of configured actions.

You can have as many lists as you require. After configuring a list, you can include it in the policy session-config, the default-session-config, or both. Lists are applied to an instant message in the order in which they are included by the session-config, and chosen by the AA-SBC policy engine for the particular session—all word lists first, followed by all URL lists.

Changes made to word or URL lists take effect immediately on all traffic through AA-SBC. However, changes to lists that are associated with policies do not take effect on IM sessions that are open at the time you made the configuration change. You must close the IM windows and re-open them for the changes to take effect. (You do not need to sign out from the IM clients.)

The word and URL list pre-stamps and post-stamps specify text added to the instant-messaging session before (pre) and after (post) the instant message that can be viewed by the recipient.

Hooking lists to policies

Once you have configured your word and URL lists, you can use them in any policy and/or the default-session-config. To do this, from the appropriate session-config object open the instant-messaging-content object. You can then set the word-list or url-list properties to include the appropriate lists using a full path name to the IM filtering lists, as follows:

```
config> vsp
config vsp> config default-session-config
config default-session-config> config instant-messaging-content
config instant-messaging-content> set url-list
   “vsp\im-filtering\url-list good-guys”
config instant-messaging-content> set url-list
   “vsp\im-filtering\url-list bad-guys”
config instant-messaging-content> set word-list
   “vsp\im-filtering\word-list bad-words”
```

IM filtering object summary

The following table lists and briefly describes the **im-filtering** objects. See the following chapter for other objects in the CLI hierarchy:

- Chapter 75, “Virtual system partition (VSP) objects”

Object name	Description
im-filtering	Opens the instant messaging filter object for configuration.
word-list	Opens or creates the specified word-list configuration object on AA-SBC. The system scans the contents of instant messages for these words.
url-list	Opens or creates the specified URL list configuration object on AA-SBC. The system scans the contents of instant messages for these URLs.

im-filtering

Purpose

Opens the instant messaging filtering object for configuration. From this object you can configure lists of words and URLs to the system scans for in instant messaging content. In addition, you can define the **url-extractor-regular-expression** property to define for AA-SBC what is considered a URL.

When AA-SBC processes a URL list, which identifies which URLs to take action on, it first needs to know what URLs are contained in the message. Using the **url-extractor-regular-expression** value as a definition, AA-SBC first extracts a list of all URLs in a message. This list of words is then compared against the list of domains/expressions defined in the url-list object.

The following string is the standard extractor regular expression:

```
\b((?i)((https?|ftp|file):[\\/][\\/
   ][\w[:punct:]]*[\w~`@#$%\^&*_\-=\\|/
   :]{1})|(((([\w_%\-]+\.)+(com|org|net|edu|biz|info|name|museum|coop
   |aero|pro|[a-z]{2}))|((\d{1,3}\.){3}\d{1,3}))([\\:/
   ]([\w[:punct:]]*[\w~`@#$%\^&*_\-=\\|/:]{1})?)?))\b
```

Syntax

```
config vsp im-filtering
```

Properties

<table>
<tr><th>Property name</th><th>Description</th></tr>
<tr><td>url-extractor-regular-expression {standard | custom <em>urlExtractorRegExp</em>}</td><td>Generically defines what should be considered a URL in scanned instant messages. This property stores a single regular expression, which the system then uses to define what is considered a URL. The system scans instant message content and extracts URLs that match this expression. It is unlikely that you would change this definition from the default setting, standard, which uses an internally defined regular expression that catches all URLs.

In the following example, the custom regular expression adds recognition of the new .jobs and .travel top-level domains that are approved by ICANN, but are not yet in operation. Note that this string would be entered on a single line.

<b>Example: “\b((?i)((https?|ftp|file):[\\/][\\/][\w[:punct:]]*[\w~`@#$%\^&*_\-=\\|/:]{1})|(((([\w_%\-]+\.)+(com|org|net|edu|biz|info|name|museum|coop|aero|pro|jobs|travel|[a-z]{2}))|((\d{1,3}\.){3}\d{1,3}))([\\:/]([\w[:punct:]]*[\w~`@#$%\^&*_\-=\\|/:]{1})?)?))\b”</b>
The default setting is <b>standard</b>.</td></tr>
</table>

word-list

Purpose

Opens the word-list object for editing. A word-list contains a list of words and/or expressions to search for in instant messaging content, and the action(s) to take when a match is found. Once configured, you can then include your word list within the instant-messaging-content object in either:

- the default-session-config object
- the session-config object
- the session-config-pool object
- the dial-plan object

See Hooking lists to policies for a sample of including a word-list.

Finding Windows-style file references

If you want AA-SBC to scan the content for Windows-style file references, you can do so by adding the following regular expression:

```
config word-list files> set entry expression
   (\\\\[\w[:punct:]]*[\w~`@#$%\^&*_\-=\\|/:])
```

This expression matches all file references in the form \\machinename\sharename\path\file. The expression uses the same rules as Windows Messenger. If a space exists in the filename, the nd of the file will not be detected.

Syntax

```
config vsp im-filtering word-list name
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables the word list in which you are working. **Example: set admin enabled** The default setting is **enabled**.
entry {word `string` | expression *`regExp`* [*`description`*] | file *fileName* {word | expression}}	Adds a word, expression, or a list of either to the list that the system matches the message content against (and takes subsequent actions on). The system accepts any number of entries. If the entry type is **word**, the system makes the following changes to the string: 1. \b is added to the beginning and end, so that only whole words are matched. 2. (?i) is added to the beginning, so that searches are case-insensitive. 35. All characters that have special meaning for the regular expression engine are “escaped” so that the literal character is searched for instead. If the entry type is **expression**, the text is interpreted as a PCRE regular expression. By default it does not check for word boundaries, but does compare case sensitivity. You can add a description to the entry to aid in deciphering the regular expression when you go back to look at it again later. Note that the expression comparison is, by default, “greedy.” This means that a search for “.*” will find the longest match possible. To turn greedy off, use (?U). *Continued*

Property name	**Description**
entry {word *string* | expression *regExp* [*description*] | file *fileName* {word | expression}}	*Continued* If the entry type is **file**, the system reads all words or regular expressions contained in the file at startup and matches against them. Files must be plain ASCII text with either DOS CRLF-terminated or Unix LF-terminated lines. Enter a full path name to the file, preferably located in the /cxc_common directory, and indicate whether it is a list of words or expressions. You can update the file contents at any time, and reread the file into memory using the file-based-word-lists-refresh action. **Example: set entry expression “\b([Cc])overgence\b” “match companyABC”** There is no default setting.

Property name	Description

replace-text {enabled [*replacementText*] | disabled}	Enables or disables replacement of words or expressions that match configured entries. If enabled, specify the text to put in place of all word or expression matches in the list. If you enable **replace-text**, but do not specify any text, the system replaces the matching word or expression with nothing.
	Note that you can use the regexp subpattern specifiers ("\1", \2", etc.) in the replacement text, assuming that you used the subpattern markers ("("and ")") in the expressions. For example, if you had a match expression of:
	`\b([Cc])ompany123\b`
	and a replacement-text of:
	`\1ompanyXYZ`
	all occurrences of "Company123" would be replaced with "CompanyXYZ" and all occurrences of "company123" with "companyXYZ".
	Example: set replace-text enabled "—TEXT BLOCKED—"
	The default setting is **disabled**.
	For more information regarding configuring regular expressions and replacement strings, see the section in Chapter 1, "Using regular expressions" on page 36.

Property name	Description
directive {allow | discard | refuse [*resultCode*] [*resultString*] | follow-sip-directive}	Assigns an action to the message. Select one of the following actions: • **allow**—allows the message, even if higher-level policy (under instant-messaging) says to refuse or discard it. • **discard**—silently deletes the message instead of delivering it. No notification is sent. • **refuse**—deletes the message, but sends a SIP error response to the sending agent. Optionally, specify the result code, between 400 and 699, and/or a result string to send in the error response. The default error code is 400, with no accompanying text. • **follow-sip-directive**—follows whatever actions are configured at the session-level. (These are the settings under sip-directive, and/or instant-messaging.) Note that if you specify the **refuse** directive with text, the text will be placed on the method line of the SIP response message. That line is usually not displayed to the user. If you want a message displayed to the sender, use the **message-to-sender** property. **Example: set directive refuse 500 “Message discarded”** The default setting is **follow-sip-directive**.

Property name	Description
alert {enabled `eventLogTarget` [`severity`] | disabled}	Enables or disables sending alert messages to an event log. If enabled, the system sends an IM alert message to the specified target. Enter the full path name to either the local database, the syslog server, or the event log file that serves as the target. An IM alert message contains the data of the From and To fields, as well as the content of the message. Optionally, you can change the severity level of the messages sent. Note that the system sends word-lists alerts in addition to any session-level alerts. One alert is sent for each word-list that had one or more match (and had an alert enabled). **Example: set alert enabled "services\event-log\file messages"** The default setting is **disabled**. If enabled, the default severity level is **alert**.
snmp-trap {enabled | disabled}	Enables or disables the system from sending an SNMP trap if a word list match is found. If enabled, traps are sent to the configured trap target (**box interface** *eth#* **ip** *intName* **snmp**). Each time that word-list is matched, the system sends a trap containing the content of the To and From fields, the name of the word list, and the action taken. **Example: set snmp-trap enabled** The default setting is **disabled**.

Property name	Description
archiving {enabled | disabled}	Writes the original and modified contents of this message to the IM archive database. If the session-level instant-messaging/archiving is already turned on, this cannot be used to turn it back off for this particular message—the two settings are OR'd together to determine the action to take. Use this option if you do not normally archive messages, but want to archive those that contain certain material. **Example: set archiving enabled** The default setting is **disabled**.
pre-stamp {enabled *text* | disabled}	Sets whether to prepend text to the message, before delivery to the recipient, if there is a match with one of the word or expression entries. Use this option, for example, to print a summary notice of what was done to the message at the beginning. Use "\r" to insert a line break in your message. **Example: `set pre-stamp enabled "NOTICE: This message was censored\r"`** The default setting is **disabled**.
post-stamp {enabled *text* | disabled}	Sets whether to append text to the end of the message, before delivery to the recipient, if there is a match with one of the word or expression entries. Use this option, for example, to print a summary notice of what was done to the message at the message end. Use "\r" to insert a line break in your message. **Example: `set post-stamp enabled "Antiulcer This message was censored"`** The default setting is **disabled**.

Property name	Description
message-to-sender {enabled *text* | disabled}	Returns a message to the originator of messages that contain any matching words or expressions.The message is prepended onto the next message through the system destined for the sender. **Example:** **`set message-to-sender enabled “NOTICE: Your last message was censored\r”`** The default setting is **disabled**.
message-to-recipient {enabled *text* | disabled}	Sends a “notification” message to the intended recipient of a message that contains any matching words or expressions. This notification is sent prior to the actual message (which may or may not be sent). **Example:** **`set message-to-recipient enabled “NOTICE: The following message contains censored material\r”`** The default setting is **disabled**.

url-list

Purpose

Opens the URL-list object for editing. A URL list contains a list of domains and/or expressions to search for in instant messaging content, and the action(s) to take when a match is found. Once configured, you can then include your URL list within the **instant-messaging-content** object in either:

- the default-session-config object
- the session-config object.

See Hooking lists to policies for a sample of including a URL list.

The URL list uses the regular expression defined in the **url-extractor-regular-expression** property in the im-filtering object to extract URLs from IM content.The extractor is a single regular expression that defines what is considered a URL. The content that matches the expression can then, for example, be virus scanned. If a URL does not match the **url-extractor-regular-expression**, you can not add an entry in the url-list that matches and finds that URL. To put it in algorithmic terms, the determination of a match is: if (match(string, url-extractor-regexp) AND match(string, url-list-item) AND (virus-check-disabled or virus-found)) then take actions.

AA-SBC compares entries sequentially against each URL that it extracts from the IM content. When you have configured multiple URL lists, AA-SBC processes them in the order you included them in the **instant-messaging-content** object.

If your URL list has an empty **entry** property, the actions defined in the list are taken for all URLs. In other words, having nothing to match means match everything. (This would happen if you created a list but did not set the **entry** property.) Because the regular expression extractor, by default, extracts all URLs, the **entry** property in essence, contains those matches. If you add an **entry** property to the url-list object, and do not supply a domain or expression to match, you no longer have an empty entry, you have an entry that matches nothing.

Syntax

```
config vsp im-filtering url-list name
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables the named URL list in which you are working. **Example: set admin enabled** The default setting is **enabled**.
entry {domain `domainName` | expression `regExp` [`description`] | file *fileName* {domain | expression}}	Adds a domain name, regular expression, or a list of either to the list that the system matches the message content against (and takes subsequent actions on). The system accepts any number of entries. If the entry type is **domain**, the system makes the following changes to the string (and then treats it like an expression): 1. .* is added to the beginning, so that the match does not have to start at the beginning of the string. 2. \b is added to the beginning and end, so that only whole words are matched. There must be a word break (a non-alphanumeric character, or a beginning end of string) before and after the domain. 36. (?i) is added to the beginning, so that searches are case-insensitive. All characters that have special meaning for the regular expression engine are “escaped” so that the literal character is searched for instead. If the entry type is **expression**, the text is interpreted as a PCRE regular expression. By default it does not check for word boundaries, but does compare case sensitivity. You can add a description to the entry to aid in deciphering the regular expression when you go back to look at it again later. By default, expressions must match starting at the first character of the URL string. You can, however, modify this using meta-characters in the expression. *Continued*

Property name	Description
entry {domain *domainName* | expression *regExp* [*description*] | file *fileName* {domain | expression}}	*Continued* Note that the expression comparison is, by default, “greedy.” This means that a search for “.*” will find the longest match possible. To turn greedy off, use (?U). If the entry type is **file**, the system reads all domain names or regular expressions contained in the file at startup and matches against them. Files must be plain ASCII text with either DOS CRLF-terminated or Unix LF-terminated lines. Enter a full path name to the file, preferably located in the /cxc_common directory, and indicate whether it is a list of domain names or expressions. You can update the file contents at any time, and reread the file into memory using the file-based-word-lists-refresh action. **Example: set entry domain “\b([Cc])overgence.com\b” “match companyABC.com”** There is no default setting.
replace-text {enabled [*replacementText*] | disabled}	Enables or disables replacement of domains or expressions that match configured entries. If enabled, specify the text to put in place of all domain or expression matches in the list. If you do not specify replacement text, the system removes the matching word or expression and does not put anything in its place. **Example: set replace-text enabled “—URL BLOCKED—“** Note that you can *not* use the regexp subpattern specifiers (“\1”, \2”, etc.) in the replacement text. The default setting is **disabled**. For more information regarding configuring regular expressions and replacement strings, see the section in Chapter 1, “Using regular expressions” on page 36.

Property name	Description
directive {allow | discard | refuse [`resultCode`] [`resultString`] | follow-sip-directive}	Assigns an action to the message containing the matching URL. Select one of the following actions: • **allow**—allows the message, even if higher-level policy (under instant-messaging) says to refuse or discard it. • **discard**—silently deletes the message instead of delivering it. No notification is sent. • **refuse**—deletes the message, but sends a SIP error response to the sending agent. Optionally, specify the result code, between 400 and 699, and/or a result string to send in the error response. The default error code is 400, with no accompanying text. • **follow-sip-directive**—follows whatever actions are configured at the session-level. (These are the settings under sip-directive, and/or instant-messaging.) Note that if you specify the **refuse** directive with text, the text will be placed on the method line of the SIP response message. That line is usually not displayed to the user. If you want a message displayed to the sender, use the **message-to-sender** property. **Example: set directive refuse 500 “A URL was deleted due to a suspected virus”** The default setting is **follow-sip-directive**.

Property name	**Description**
alert {enabled `eventLogTarget` [`severity`] | disabled}	Enables or disables sending alert messages to an event log. If **enabled**, the system sends an IM alert message to the specified target. Enter the full path name to either the local database, the syslog server, or the event log file that serves as the target. An IM alert message contains the data of the From and To fields, as well as the URL that the system matched on. Optionally, you can change the severity level of the messages sent. One alert will be sent for each match in each URL list. This is in addition to any alerts sent as a result of other configurations. **Example: set alert enabled "services\event-log\file messages"** The default setting is **disabled**. If enabled, the default severity level is **alert**.
snmp-trap {enabled | disabled}	Enables or disables the system from sending an SNMP trap if a URL list match is found. If enabled, traps are sent to the configured trap target (**box interface** *eth#* **ip** *intName* **snmp**). The system sends one trap for each url-list match. This is in addition to any traps sent for word-list matches. **Example: set snmp-trap enabled** The default setting is **disabled**.
archiving {enabled | disabled}	Writes the original and modified contents of this message to the IM archive database. If the session-level instant-messaging/archiving is already turned on, this cannot be used to turn it back off for this particular message – the two settings are OR'd together to determine the action to take. Use this option if you do not normally archive messages, but want to archive those that contained certain URLs. **Example: set archiving enabled** The default setting is **disabled**.

Property name	**Description**
pre-stamp {enabled *text* | disabled}	Sets whether to prepend text to the message, before delivery to the recipient, if there is a match with one of the domain or expression entries. Use this option, for example, to print a summary notice of what was done to the message at the beginning. Use “\r” to insert a line break in your message. **Example:** **`set pre-stamp enabled “NOTICE: A URL contained in this message was censored\r”`** The default setting is **disabled**.
post-stamp {enabled *text* | disabled}	Sets whether to append text to the end of the message, before delivery to the recipient, if there is a match with one of the domain or expression entries. Use this option, for example, to print a summary notice of what was done to the message at the message end. Use “\r” to insert a line break in your message. **Example:** **`set post-stamp enabled “NOTICE: A URL contained in this message was censored\r”`** The default setting is **disabled**.
message-to-sender {enabled *text* | disabled}	Returns a message to the originator of messages that contain any matching domains or expressions.The message is prepended onto the next message through the system destined for the sender. **Example:** **`set message-to-sender enabled “NOTICE: Your last message was censored\r”`** The default setting is **disabled**.
message-to-recipient {enabled *text* | disabled}	Sends a “notification” message to the intended recipient of a message that contains any matching domains or expressions. This notification is sent prior to the actual message (which may or may not be sent). **Example:** **`set message-to-recipient enabled “NOTICE: The following message contains censored material\r”`** The default setting is **disabled**.

IM Filtering objects

35. IP objects

IP description

This chapter describes the Internet Protocol (Version 4) configuration objects in AA-SBC. IP objects are identified by a unique string name. By using names, you can change the underlying IP address and mask without disrupting the interface. This means that you do not need to first delete an interface if you need to edit the address/ mask. Interfaces that boot using the Dynamic Host Configuration Protocol (DHCP) also use the string name, since there is no IP address or mask for the interface.

IP object summary

The following table lists and briefly describes the **ip** objects. See the following chapters for other objects in the CLI hierarchy:

- Chapter 14, “Cluster, box, and interface objects”
- Chapter 77, “VLAN objects”

Object name	Description
ip	Configures parameters of IP on Ethernet and VLAN interfaces.
telnet	Configures the parameters of the Telnet session. See Chapter 69, “Telnet objects” for information.
ssh	Configures the parameters for secure client/ server communications, remote logins, and file transfers using encryption and public-key authentication. See Chapter 58, “Secure Shell objects” for information.

Object name	Description
`snmp`	Configures remote management access using the Simple Network Management Protocol. See Chapter 66, “SNMP objects” or information.
`web`	Enables the Web server, providing access to the AA-SBC graphical user interface. See Chapter 79, “Web objects”, for information.
`web-service`	Configures the system as a web service server. See Chapter 80, “Web-service objects”, for information.
`eventpush-service`	Configures and redirects AA-SBC logged events to an external Web browser. See Chapter 28, “Eventpush service objects”, for information.
`ipsec-tunnel`	Configures IPSec tunnels on the interface. See Chapter 36, “IPSec objects”, for information.
`ipsec-transport`	Configures an IPSec policy in transport mode. See Chapter 36, “IPSec objects”, for information.
`ike`	Configures Internet Key Exchange (IKE) on the IP interface. See Chapter 36, “IPSec objects”, for information.
`sip`	Configures the Session Initiation Protocol (SIP), described by RFC 3261. See Chapter 65, “Session Initiation Protocol objects”, for information.

Object name	Description
`h323`	Configures an interface for H.323, the recommendation from the ITU Telecommunication Standardization Sector, providing audio-visual communication sessions on packet networks. See Chapter 32, “H.323 objects”, for information.
`ntp-server`	Configures Network Time Protocol client and server parameters. See Chapter 44, “NTP client and server objects”, for information.
`tftp`	Configures the TFTP server to upload and download executable images and configurations between the system and other devices. See Chapter 70, “TFTP server objects”, for information.
`bootp-server`	Configures the Bootstrap Protocol (BOOTP) client and server settings in a system network cluster. See Chapter 10, “BOOTP client and server objects”, for information.
`icmp`	Configures the Internet Control Message Protocol (ICMP), which determines whether a destination is unreachable. See Chapter 33, “ICMP object”, for information.
`vrrp`	Configures the interface to be available to the Virtual Router Redundancy Protocol (VRRP), which provides link-level failover capabilities between two or more virtual interfaces. See Chapter 78, “VRRP objects”, for information.

Object name	Description
`media-ports`	Configures the IP address and port range to assign to SIP traffic originating on the private side of the system network. See Chapter 40, “Media ports object”, for information.
`near-side-nat`	Configures the system to perform address translation on behalf of an enterprise firewall device. See Chapter 43, “Near-side NAT object”, for information.
`routing`	Manually creates static IP routes to destination networks and hosts (routers) connected to the Internet. See Chapter 57, “Routing objects”, for information.
`dns-server`	Identifies the IP interface on which the DNS server resides. See Chapter 25, “DNS service resolver and server objects”, for information.
`stun-server`	Configures the system as a STUN server to identify the public-side NAT details by inspecting exploratory messages from STUN-enabled clients. See Chapter 68, “STUN server objects”, for information.
`proxy`	Configures a generic proxy interface on the system, allowing communication between devices that are each behind a firewall or NAT. See Chapter 53, “Proxy interface object”, for information.
`kernel-filter`	Creates allow and deny rules which the system processes at the kernel level, before higher level, more compute intensive rules. See Chapter 37, “Kernel filter rule objects”, for information.

Object name	Description
messaging	Configures a listening socket on an IP interface. This enables the interface to receive messaging traffic and participate in clustering and media partnering. See Chapter 41, “Messaging objects”, for information.
diameter	Configures the Diameter protocol on the system, allowing it to act as Diameter client and/or server. See Chapter 22, “Diameter client and server objects”, for information.

ip

Purpose

Opens the named IP configuration object for editing. Specify the name of the IP interface using up to 16 alphanumeric characters with no blank spaces. If you intend for the interface to be a headend interface to support load-balancing of SIP processing, see Configuring head-end and backing interfaces for more information.

Tag-based route selection

AA-SBC uses classification tags to classify incoming traffic and routing tags to control the egress route for a specific service type. This may be useful, for example, in E911 applications. With inbound traffic to AA-SBC on an interface, you may want to ensure that it always goes out on a specific interface. To do this, you would configure a classification-tag on the incoming interface that matches the routing-tag on the egress interface you desire.

When you configure an IP interface, AA-SBC installs both a network route and a host route into the generic routing table. For example, suppose you create an IP interface named ABC with static IP address 1.1.1.1/32:

```
Generic route table for ABC
-------------------
1.1.1.1/32
1.1.1.0/24
```

If there are services configured under the interface (i.e., media, SIP, or STUN), the route is also installed in the specific service routing table. (See Services routing description for a general description of service route tables.) For example, if you configured SIP on interface ABC:

```
Generic route table for ABC
-------------------
1.1.1.1/32
1.1.1.0/24

SIP service route table for ABC
-------------------
1.1.1.1/32
1.1.1.0/24
```

However, if you create a routing-tag for an interface, AA-SBC creates a separate service route table for that tag, populated with any static routes configured on that interface. When the first routing-tag is configured, AA-SBC removes the routes associated with that interface from the default service route table. They are only available in the service route tables associated with the routing-tag(s).AA-SBC does not install (or removes) interfaces that have a routing-tag applied from the default service routing table. For example, if you created routing-tag E911 on interface ABC:

```
Generic route table for ABC
-------------------
1.1.1.1/32
1.1.1.0/24

SIP service route table for ABC
-------------------

SIP service route table for ABC.E911
-------------------
1.1.1.1/32
1.1.1.0/24
```

To retain the route in both the default service route table and the tag-specific service route table, add a **routing-tag** named “null.” This reserved routing-tag name indicates that the service routes should be installed in the default service route table as well.

Note that tag-based service route tables inherit the metrics assigned to that service type with the services-routing metrics. In addition, if a matching session configuration includes a routing-settings ingress-classification-tag for incoming traffic, the session config setting takes precedence. See the **routing-tag** and **classification-tag** descriptions in this object for specific configuration requirements.

Note: The preferred method for creating virtual firewalls is by using routing tags and VLANs. For sample configurations that illustrate VLANs, overlapping IP addresses, and virtual firewalls, see the *Net-Net OS-E – System Administration Guide*.

Syntax

```
config cluster box number interface ethX ip name
config cluster box number interface ethX vlan integer ip name
config cluster vrrp vinterface vxID ip name
config cluster vrrp vinterface vxID vlan integer ip name
config box interface ethX ip name
config box interface ethX vlan number ip name
```

Properties

Property name	**Description**
admin {enabled | disabled}	Enables or disables IP services on this interface. **Example: set admin enabled** The default setting is **enabled**.
ip-address {DHCP | static *ipaddress/ mask*}	Sets Dynamic Host Configuration Protocol (DHCP) IP address assignment on this interface from a DHCP server, or sets a static IP address and network mask. For static IP addresses, specify the IP address and network mask for this Ethernet interface. **Example: `set ip-address dhcp`** **`set ip-address static 192.67.43.4/32`** The default setting is **dhcp**.
geolocation *integer*	Assigns a numeric to the IP interface that you can later use, for example, within a policy to identify traffic to or from that interface. To use the policy match feature, set the session configuration routing-settings object. You can also use this value as a filtering mechanism with the service-route-lookup action to return the best route to a destination. **Example: set geolocation 10** The default setting is **0**.

Property name	Description
metric *preference*	Associates a cost with the interface routes (both host and network routes) that the system adds to its services route and route DB tables. The system chooses the more preferred route when there are multiple interfaces available on the same network. The lower the metric the more preferred the route. This value is carried over to the VSP services-routing metrics as the **user-metric** value. **Example: set metric 10** Enter a value between 0 and 4294967295; the default setting is **1**.
classification-tag *string*	Associates the classification-tag with the incoming service on this interface. The classification-tag applies to the ingress interface over which the system initially receives service traffic. Each IP interface can have at most one classification tag. This tag must match a configured **routing-tag** for tag-based route selection to be in effect. You can also configure ingress or egress classification tags through the session-config routing-settings object. If this property is configured in both places, the routing-settings configuration takes precedence. Note that this tag is case-sensitive. **Example: set classification-tag E911** There is no default setting.

Property name	Description
routing-tag *string*	Associates all the routes configured on an interface with this routing-tag and creates a service route table based on the routing-tag for each service enabled on this interface. The routing-tag applies to the egress interface over which the system forwards service traffic. In order to perform tag-based routing, a classification-tag must be configured on the ingress interface over which the system initially receives service traffic, and that classification tag must match the routing-tag. Each IP interface can have multiple routing tags. (Classification tags in the session-config routing-settings object also must match this routing tag set in the ip object. Once a routing-tag is configured for an interface, the service routes associated with that interface are installed in the service route table associated with the routing-tag(s). In other words, the service routes are no longer installed in the default service route tables—they are only in the service route tables specified by the routing-tag(s). However, in order for tag routing to be in effect for media, the **tag-routing** property of the matching session config media object must be enabled (it is disabled by default). If you create an additional routing-tag for the interface with the name “null,” the system installs the route in both the default service route table and the tag-specific service route table. Note that this tag is case-sensitive. **Example: set routing-tag E911** There is no default setting.
security-domain {trusted | untrusted}	Sets the informational text string that indicates the trust level of this IP interface. For example, interfaces that point to the internal network are **trusted**; interfaces that point to the public DMZ-side of the network are **untrusted**. You can then use this setting to identify an interface within your policy configuration. **Example: set security-domain untrusted** There is no default setting.

Property name	Description
trusted-peer *ipAddress*	Configures one or more trusted peers for this IP interface. The system accepts and processes all TCP traffic received from a trusted peer. Use this property to designate servers as trusted peers in a VRRP topology that uses TCP as the transport between the system and the server. If a failover should occur, the backup system will accept server traffic and send a TCP reset to close the connection to the failed system and establish a new one for itself. **Example: set trusted-peer 10.10.10.1** There is no default setting.

Property name	Description
address-scope {public | private}	Sets the informational text string that indicates the private or public scope of this IP interface. For example, interfaces with private IP addresses on the internal network can be configured as **private**; interfaces with public IP addresses to the external network can be tagged as **public**. You can then use this setting to identify an interface within your policy configuration. **Example: set address-scope private** There is no default setting.
filter-intf {enabled | disabled}	Enables or disables secure traffic filtering on this IP interface. When enabled, inbound packets that match one of the configured IP addresses on this interface are allowed to pass. All other IP packets are blocked. This enforces the concept that packets destined for an interface must actually come in on that interface. For example, consider a box with two Ethernet interfaces—1.1.1.1 on interface A and 2.2.2.2 on interface B. When disabled, pinging either address from the B side of the network will succeed, even though 1.1.1.1 is an A-side IP address. However, when **filter-intf** is set to **enable**, pinging 1.1.1.1 from the B side fails. Set this to enabled to add another level of security to AA-SBC, however, make certain that you fully understand your network structure before setting up this traffic filtering. **Example: set filter-intf enabled** The default setting is **disabled** and all IP packets are allowed to pass.

36. IPSec objects

IPSec description

IPSec is a combination of Internet protocols designed to protect internet communications at the IP layer. IPSec is comprised of two major concepts—security policies (SP) and security associations (SA). It uses the Internet Key Exchange (IKE) for authenticating peers, exchanging shared secret keys, and setting up SAs.

Security policies (SP)

SPs are used to filter packets and determine how to process them (encrypting, decrypting, pass through, or drop). They are configured on specific IP interfaces, and are stored in the Security Policy Database (SPD). AA-SBC SPs usually specify the following parameters:

- The source and destination IP addresses of the packets to be protected. In transport mode these are the same addresses as those in the SA. In tunnel mode, they may be different. (See below for a description of transport and tunnel.)
- The IPSec protocol used to protect packets.
- Policy-related behaviors for transport connections.

When SPs are configured on a AA-SBC IP interface, the system uses either transport or tunnel mode. Transport mode is used for hosts that need to communicate with the system directly and securely. Tunnel mode is used for communication through security gateways. For example, if there are a group of trusted hosts in a LAN, they can communicate with each other without security (within the trusted LAN). If one should need to communicate with the AA-SBC device outside the LAN, it would do so through a security gateway. The connection between the gateway and the system is secured by an IPSec tunnel. There is no IPSec requirement on each host, but the traffic between AA-SBC and the hosts would travel through the secure IPSec tunnel.

Security associations (SAs)

SAs describe how two entities will communicate securely with one another. SAs are created by IKE and are stored in the Security Association Database (SAD). The AA-SBC uses the Internet Key Exchange (IKE) to automatically negotiate SAs between the two IPSec participants. A security association protects in one direction when two peers are communicating; two SAs are required to protect bidirectional traffic between peers. Each SA has the following parameters:

- Source and destination IP addresses of the resulting IPSec header. (These are the IP addresses of the IPSec peers.)
- IPSec protocol (AH or ESP).
- Algorithm and secret key used by IPSec protocol.
- Encryption and authentication algorithms.
- Diffie-Hellman group identifiers.
- Valid lifetime timer.

Internet Key Exchange (IKE)

IKE manages peer authentication and the exchange of symmetric keys. It then creates the SAs and adds them to the SAD. IKE also periodically rekeys the secret keys to ensure their confidentiality. Typically using UDP port 500 for communication, the IKE protocol functions in two phases:

1. Phase 1 establishes an Internet Security Association Key Management Protocol Security Association (ISAKMP SA) to authenticate the peers. This can be done using pre-shared keys (PSK), RSA keys, and/or X.509 certificates. There are main, base, and aggressive modes. Main mode requires six packets but is completely secure. Base mode uses four-packet exchange and provides protection against denial of service attacks. Aggressive mode uses half as many messages as main mode, but does not support identity protection (because some information is passed in clear text) and is susceptible to “man in the middle” attacks.
2. In Phase 2, the ISAKMP SA is used to negotiate and setup the IPSec SAs. Usually, peers negotiate one ISAKMP SA and that is then used to create two unidirectional SAs.

If a public key method is used for IPSec authentication, each IPSec endpoint must have a public and a private key. The private key is only for the local IPSec endpoint. The public key is to be installed at IPSec endpoints where it will connect with IPSec. IKE is configured at the cluster level for the box and can be administratively managed (enabled and disabled) at the IP interface level.

Cluster IPSec objects summary

You configure IPSec cluster-wide key exchange criteria.

The following table lists and briefly describes the **cluster** IPSec objects. See the following chapters for other objects in the CLI hierarchy:

- Chapter 14, “Cluster, box, and interface objects”

Object name	Description
`ike`	Configures Internet Key Exchange parameters for the cluster
`remote`	Configures an IKE remote peer and the IKE phase 1 negotiation parameters.
`sa-info`	Configures the IKE phase 2 security association parameters for remote peers.

Interface IPSec objects summary

You configure IPSec tunnels or transports on an IP interface.

The following table lists and briefly describes the IP interface IPSec objects. See the following chapters for other objects in the CLI hierarchy:

- Chapter 14, “Cluster, box, and interface objects”
- Chapter 77, “VLAN objects”
- Chapter 35, “IP objects”

Object name	Description
ipsec-tunnel	Configures an IPSec tunnel between AA-SBC and a security gateway.
ipsec-transport	Configures an IPSec transport between AA-SBC and another host or server.
ike	Enables IKE on an interface.

ike

Purpose

Configures Internet Key Exchange (IKE) parameters for the cluster. See Internet Key Exchange (IKE) for more information. Once IKE has been configured and enabled on a cluster, you can then administratively control it on an interface-by-interface basis.

Creating a PSK file

AA-SBC can use a pre-shared key file as the authentication method for phase 1 negotiation. (You specify the method with the **auth-method** property of the remote object.) You must create this PSK file from within AA-SBC. The recommended storage location for the pre-shared key file is in the /cxc/certs directory. The pre-shared key file is organized in columns. The first column holds the identity of the peer authenticated by the pre-shared key. The second column contains the contents of the pre-shared key. To create this file, you must exit to the shell. (To do so, the **debug** property of the access permissions object must be enabled.) Create a file using the following format:

```
# IPv4/v6 addresses
1.1.1.1                 a14367af96923961294f3d7392a49689
1.1.2.1                 2154789eadb5f814943789a4a823b7ba

# USER_FQDN
jdoe@companyABC.com     janedoekey

# FQDN
companyABC.com          companyABC
```

After creating the file, you must restrict access to the file, allowing only the owner to modify it. To set permissions, shell out of AA-SBC and set access. For example, if the file were named psk.txt:

```
bash-3.00# chmod 0600 psk.txt
```

Syntax

```
config cluster ike
```

Properties

Property name	**Description**
admin {enabled | disabled}	Enables or disables IKE for the cluster. You can set the administrative state on an interface-by-interface basis using the interface ike object. **Example: set admin enabled** The default administrative state is **enabled**.
psk-file *fileName*	Specifies the path to the pre-shared key file to be used during IKE authentication. See Creating a PSK file for important information about this file. **Example: set psk-file /cxc/certs/psk.txt** There is no default setting.
cert-path *certPath*	Specifies the path to the location of the certificate file used for IKE authentication. Enter the selected certificate file type (and name, if applicable) using the remote object **cert-type** and/or **peers-cert** properties. These certificate files should reside in the **cert-path** directory. **Example: set cert-path /cxc/certs** There is no default setting.
log {notify-print | notify | debug-print | debug2-print | debug | debug2}	Sets the IKE log level. For the ability to view the messages, you must configure the local-database filter to **general** with a severity level of **info**. **Example: set log debug** The default setting is **notify**.

remote

Purpose

Configures an IKE remote peer and the IKE phase 1 negotiation parameters. These data protection parameters comprise the IKE proposal. If the AA-SBC is the connection initiator, it sends a proposal to the remote endpoint, containing the parameters for protecting the IKE connection. The responder then chooses the most suitable proposal based on its own security policies and responds with the selection. If no suitable proposal is found, the connection fails.

Syntax

```
config cluster ike remote name
```

Properties

Property name	Description
remote-peer {anonymous | address *ipAddress*}	Identifies the remote IKE peer, either by specific address or anonymously. The **anonymous** setting matches any peer that is not specifically identified by its address. **Example: set remote-peer 1.1.1.1** The default setting is **anonymous**.
ike-mode {main | aggressive | base | main_aggressive}	Configures the IKE exchange mode used in phase 1. This is the mode AA-SBC uses when it is the initiator or the responder in a connection. Select one of the following modes: • **main**—provides identity protection by requiring knowledge of the peer's pre-shared key prior to the knowledge of the peer identity. Main mode requires a six-packet exchange but is completely secure during establishment of the connection. • **aggressive**—identities appear in the first two messages. This mode requires fewer messages to accomplish authentication, but is less secure. • **base**—sends key exchange payload with authentication data for the initiating peer. • **main_aggressive**—sets the system to use main mode when the initiator and to accept main or aggressive when the responder **Example: set ike-mode main_aggressive** The default setting is **main**.

Property name	Description
my-identifier {address | fqdn *name* | user-fqdn name| asn1dn}	Specifies the identifier sent to the remote host. Select one of the following to be used for identification: • **address**—uses the local IP address of the interface that has the matching security policy. • **fqdn**—uses a fully qualified domain name. Enter the domain name, for example, companyABC.com. • **user-fqdn**—uses a fully qualified domain name with the user portion. Enter, for example, jdoe@companyABC.com. • **asn1dn**—uses an ASN.1 distinguished name. If you select this option, you must also set the **cert-type** property because the name is derived from the domain name in the certificate Subject field. **Example: set my-identifier fqdn site1@company.com** The default setting is **address**.
peers-identifier {none | address | fqdn *name* | user-fqdn name| asn1dn}	Configures the type of identification expected from the peer. To verify the peer identity, the **verify-identifier** property must be set to on. Set AA-SBC to verify peer identity based on: • **none**—AA-SBC does not verify the peer identity. • **address**—the remote IP address of the interface that has the matching security policy. • **fqdn**—a fully qualified domain name. Enter the domain name, for example, companyABC.com. • **user-fqdn**—a fully qualified domain name with the user portion. Enter, for example, jdoe@companyABC.com. • **asn1dn**—an ASN.1 distinguished name. If you select this option, you must also set the **peers-cert** property because the name is derived from the domain name in the certificate Subject field. **Example: set peers-identifier asn1dn** The default setting is **none**.

Property name	Description
verify-identifier {off | on}	Specifies whether to verify the peer-identifier. When set to **on**, the peer-identifier is compared against the ID that the peer sends in the IKE phase 1 negotiation. If the identifiers do not match, the negotiation fails. Note that if **auth_method** is set to **pre_shared_keys**, the peer identifier must also be present in the psk.txt file (along with the pre-shared key). For example, if the **peers-identifier** is set to **fqdn** with companyABC.com as the string, the psk.txt file should have an entry similar to the one below: companyABC.com 9e2237acca969bc3b94afd82e7298633r: **Example: set verify-identifier on** The default setting is **off**.
cert-type {x509 *certFile keyFile* | none}	Specifies the local certificate type and private key to be used for RSA authentication. Select either: • **x509**—an X.509 certificate, supplying the names of the certificate and secret key files. The path to the file location must be set in the ike **cert-path** property. • **none**—no certificate is in use. **Example: set cert-type x509 nnos-e.public nnos-e.private** The default setting is **none**.
peers-cert {x509 *certFile* | none}	Specifies the certificate type to expect from the remote peer. Select either: • **x509**—an X.509 certificate, supplying the name of the certificate file. The path to the file location must be set in the ike **cert-path** property. • **none**—no certificate is in use. **Example: set peers-cert x509 remote.public** The default setting is **none**.

Property name	Description
proposal-check {obey | strict | claim | exact}	Sets the checking logic to apply to the lifetime property in phase 1 and the lifetime length, key length, and PFS in phase 2. Select either: • **obey**—if AA-SBC is the responder, it uses the value sent by the initiator. • **strict**—if the responder lifetime is longer or if key length is shorter than that of the initiator, the responder uses the initiator values. Otherwise, the proposal is rejected. • **claim**—if the responder lifetime is longer or if key length is shorter than that of the initiator, the responder uses the initiator values. If the responder's lifetime length is shorter than the initiator's, the responder uses its own length and notifies the initiator. • **exact**—lifetimes must be the same or the proposal is rejected. **Example: set proposal-check obey** The default setting is **strict**.
support-proxy {on | off}	Specifies whether to support using a proxy gateway to tunnel traffic of multiple hosts through a single encrypting host. If this value is set to **on**, the value of ID payloads (in phase 2 exchanges) are used as the addresses of the endpoint SAs. **Example: set support-proxy on** The default setting is **off**.
nat-traversal {on | off | force}	Enables NAT-Traversal on AA-SBC. NAT-T allows AA-SBC or the remote peer to reside behind a NAT gateway. Select one of the following options: • **on**—NAT-T is allowed and used when a NAT gateway is detected between the peers. • **off**-NAT-T is not proposed or accepted. • **force**—NAT-T is used regardless of whether or not a NAT gateway is detected between the peers **Example: set nat-traversal force** The default setting is **off**.

Property name	Description
nonce-size *octets*	Configures IKE nonce value, in octets. A nonce value is a number, used only once, as part of the authentication protocol. **Example: set nonce-size 32** Enter a value between 8 and 256; the default setting is **16**.
dpd-delay *seconds*	Activates dead peer detection (DPD) and sets the time, in seconds, allowed between keepalive requests. A value of 0 disables DPD monitoring, but still negotiates DPD support. **Example: set dpd-delay 5** The default setting is **0** (disabled).
dpd-retry-delay *seconds*	Specifies the number of seconds to wait for a keepalive response before considering the peer failed and sending another request. This value is only applicable if the **dpd-delay** setting is a non-zero value. **Example: set dpd-retry-delay 5** The default setting is **5**.
dpd-max-fail *integer*	Specifies the maximum number unanswered keepalive requests allowed before the system considers the peer dead. This value is only applicable if the **dpd-delay** setting is a non-zero value. **Example: set dpd-max-fail 3** The default setting is **5**.
lifetime *seconds*	Specifies a security association lifetime (in seconds) to propose in phase 1 of negotiations. This is the length of time an ISAKMP-SA stays valid before it must be renegotiated. A value of 0 disables the expiration timer. **Example: set lifetime 86400** The default setting is **0**.

Property name	Description
encrypt-algo {trides | des | aes | blowfish | cast128}	Specifies the IKE encryption algorithm to be used for phase 1 negotiations. Select either: • **trides**—uses the triple data encryption standard (DES). • **des**—uses standard DES encryption (for backward compatibility). • **aes**—uses the Advanced Encryption Standard. • **blowfish**—uses the blowfish symmetric key block cipher standard. • **cast128**—uses the CAST-128 block cipher. **Example: set encrypt-algo aes** The default setting is **trides**.
hash-algo {md5 | sha1 | sha256 | sha384 | sha512}	Defines the hash algorithm to be used for the phase 1 negotiations. Select either: • **md5**—uses the Message Digest 5 (MD5) algorithm. • sha algortihms—uses the Secure Hash Algorithm, resulting in a 160- (for sha1), 256, 384, or 512-bit message digest. **Example: set hash-algo md5** The default setting is **sha1**.
auth-method {pre_shared_key | rsasig}	Defines the authentication method used for the phase 1 negotiation. Select either: • **pre_shared_key**—uses PSK (shared secret). • **rsasig**—used for plain RSA authentication **Example: set auth-method pre_shared_key** The default setting is **rsasig**.
dh-group {1 | 2 | 5 | 14 | 15 | 16 | 17 | 18}	Specifies the Diffie-Hellman group identifier. When using aggressive mode, you must define the same DH group in each proposal. **Example: set dh-group 14** The default setting is **2**.

sa-info

Purpose

Configures the parameters of IKE phase 2 negotiation for establishment of an IPSec-SA. AA-SBC matches these parameter profiles against the security policies configured for an IP interface. This matching is done using the saID property of this object against the addresses configured for the security policy. (The sa-info profile that gets applied to a connection is the one that matches the security policy addresses.) For example, if you configure an ipsec-tunnel with the **from** property set to 1.1.1.1/32 and the **to** property set to 1.1.1.253/32, the local and remote addresses configured for the saID should match.

Syntax

```
config cluster ike sa-info name
```

Properties

Property name	Description
saID {anonymous | address *localIP/ mask remoteIP/mask*}	Configures the matching criteria to be used by AA-SBC in determining the sa-info profile to apply to a connection. If you set **saID** to **address**, specify the address and mask for the local and remote ends of the security association. These are the same endpoints identified in the ipsec-tunnel or ipsec-transport configuration. A cluster should have only one sa-info profile set to **anonymous**. AA-SBC uses the anonymous profile if a more specific match is not found. **Example: set saID 1.1.1.1/32 2.2.2.2/32** There is no default setting.
pfs-group {1 | 2 | 5 | 14 | 15 | 16 | 17 | 18}	Specifies the Diffie-Hellman group identifier used for Perfect Forward Secrecy (PFS). PFS ensures that each SA key is derived from a unique secret, preventing attacker from deriving additional keys if one is compromised. **Example: set pfs-group 1** The default setting is **2**.
encrypt-algo {trides | des | aes | twofish | blowfish | null-enc}	Configures the encryption algorithm used in phase 2, the negotiation and setup of the IPSec-SA. **Example: set encrypt-algo trides** The default setting is **trides**.
auth-algo {hmac-sha1 | hmacs-md5 | hmac-sha256}	Configures the authentication algorithm used in phase 2 negotiation of IPSec-SA establishment. **Example: set auth-algo hmacs-md5** The default setting is **hmac-sha1**.

Property name	Description
compress-algo deflate	Configures the compression algorithm for IP payload compression protocol (IPComp). **Example: set compress-algo deflate** The default setting is **deflate**.
lifetime *seconds*	Configures the IPSec SA's lifetime, in seconds. This is the amount of time that the authentication for the association is valid. By setting this lifetime value shorter than the lifetime value used in phase 1 negotiations (with the remote object), the phase 2 IPSec connection negotiations can be periodically re-established without having to re-do phase 1 negotiations. A value of 0 disables the expiration timer. **Example: set lifetime 28800** The default setting is **0**.

ipsec-tunnel

Purpose

Configures an IPSec tunnel for this IP interface. This object creates a tunnel between the AA-SBC and a security gateway. Tunnel mode encapsulates the entire IP packet inside another and sends it to the destination over a secure tunnel, forming a VPN over an untrusted network. For example, networks A and B are communicating over an untrusted network (e.g., the Internet). By creating an IPSec tunnel, the packets traveling over the untrusted network are protected, thereby creating a VPN. All the devices on Network A and Network B can communicate securely over the IPSec tunnel. Note that for IPSec tunnel mode for a virtual firewall interface, the IPSec tunnel must be configured on the virtual firewall's private IP interface.

Syntax

```
config cluster box number interface ethX ip name ipsec-tunnel name
config cluster box number interface ethX vlan number ip name
   ipsec-tunnel name
config box interface ethX ip name ipsec-tunnel name
config cluster vrrp vinterface vxX ip name ipsec-tunnel name
config cluster vrrp vinterface vxX vlan number ip name ipsec-tunnel
   name
```

```
config box interface ethX vlan number ip name ipsec-tunnel name
```

Properties

Property name	Description
remote-address ipAddress	Identifies the remote endpoint of the IPSec tunnel. **Example: set remote-address 1.1.1.1** There is no default setting.
from *ipAddress/mask*	Identifies the subnet or host from which the traffic originates. **Example: set from 2.2.2.2/32** There is no default setting.
to *ipAddress/mask*	Identifies the subnet or host to which the traffic is destined. **Example: set to 3.3.3.0/24** There is no default setting.

Property name	Description
proto {ah | esp}	Specifies the IPSec protocol used to protect the traffic being transported through the tunnel. Select either: • **ah**—the authentication header protocol provides data integrity and origin authentication by protecting IP payload and non-mutable header fields. Use this option when you do not need to encrypt data, but want to authenticate the remote end of the connection. (AH ensures that you are communicating with the correct remote node.) AH also detects alteration of data while in transit. • **esp**—encapsulating security payload protects the data and header of the entire IP packet providing authentication, integrity, and confidentiality. **Example: set proto ah** The default setting is **esp**.
ip-compression {enabled | disabled}	Sets whether IP payload compression is applied to packets traversing the tunnel. Compression is applied on top of the selected IPSec protocol (either **esp** or **ah**). It reduces the size of IP datagrams, increasing performance between AA-SBC and the remote node. Compression is particularly useful when encryption is applied to the packets (when **esp** is selected in the **proto** property): **Example: set ip-compression enabled** The default setting is **disabled**.

ipsec-transport

Purpose

Configures IPSec transport connection for this IP interface. Transport mode is used to protect a connection between two hosts, for example the connection between the AA-SBC and a SIP server. In transport mode the IP payload is encapsulated between the two hosts (the data below the IP header).

Syntax

```
config cluster box number interface ethX ip name ipsec-transport name
config cluster box number interface ethX vlan number ip name
   ipsec-transport name
config box interface ethX ip name ipsec-transport name
config cluster vrrp vinterface vxX ip name ipsec-transport name
config cluster vrrp vinterface vxX vlan number ip name ipsec-transport
   name
config box interface ethX vlan number ip name ipsec-transport name
```

Properties

Property name	Description
remote-address ipAddress	Specifies the far end of the transport connection. **Example: set remote-address 1.1.1.1** There is no default setting.
direction {in | out | both}	Specifies whether the transport connection can be used by inbound traffic, or outbound traffic, or both. **Example: set direction in** The default setting is **both**.
policy {discard | none | ipsec}	Specifies the action to take on traffic matching this transport connection. Select either: • **discard**—the system discards the packet. • **none**—the system does no IPSec processing on the packet. • **ipsec**—the system applies the IPSec configuration to the packet for processing. **Example: set policy discard** The default setting is **ipsec**.

Property name	Description
level {use | require}	Specifies when to apply the IPSec transport policy, depending on available security associations (SAs). Select either: • **use**— the kernel uses an SA if one is available (i.e., sends encrypted packets). Otherwise, the kernel performs normal operation (i.e., sends packets in the clear). • **require**—when a packet matches this IPSec-transport policy, an SA is required. If an SA is not present, the kernel signals IKE to establish one. **Example: set level use** The default setting is **require**.

Property name	Description
proto {ah | esp}	Specifies the IPSec protocol used to protect the traffic being transported through the transport connection. Select either: • **ah**—the authentication header protocol provides data integrity and origin authentication by protecting IP payload and non-mutable header fields. Use this option when you do not need to encrypt data, but want to authenticate the remote end of the connection. (AH ensures that you are communicating with the correct remote node.) AH also detects alteration of data while in transit. • **esp**—encapsulating security payload protects the data and header of the entire IP packet providing authentication, integrity, and confidentiality. **Example: set proto ah** The default setting is **esp**.
ip-compression {enabled | disabled}	Sets whether IP payload compression is applied to packets traversing the transport connection. Compression is applied on top of the selected IPSec protocol (either **esp** or **ah**). It reduces the size of IP datagrams, increasing performance between AA-SBC and the remote node. Compression is particularly useful when encryption is applied to the packets (when **esp** is selected in the **proto** property): **Example: set ip-compression enabled** The default setting is **disabled**.

ike

Purpose

Administratively manages IKE support for the active IP interface.

Syntax

```
config cluster box number interface ethX ip name ike
config cluster box number interface ethX vlan number ip name ike
config box interface ethX ip name ike
config cluster vrrp vinterface vxX ip name ike
config cluster vrrp vinterface vxX vlan number ip name ike
config box interface ethX vlan number ip name ike
```

Properties

Property name	Description
ike {enabled | disabled}	Enables or disables IKE messages for this interface. **Example: set ike enabled** The default setting is **enabled**.

37. Kernel filter rule objects

Kernel filter description

Kernel filter rules provide a security mechanism that allows or denies inbound traffic on AA-SBC IP interfaces. The filter controls access to resources on the enterprise servers based on source IP address and/or subnet, source port, and protocol. When AA-SBC processes kernel rules, it first interprets deny rules, then allow rules. Therefore, you can deny a subnet access, and then allow specific endpoints.

AA-SBC acts on kernel rules before the other, higher level rules such as DOS policy rules. This stops traffic from known problems early, tying up fewer processing resources.

Kernel filter object summary

The following table lists and briefly describes the **kernel-filter** objects. See the following chapters for other objects in the CLI hierarchy:

- Chapter 14, “Cluster, box, and interface objects”
- Chapter 77, “VLAN objects”
- Chapter 35, “IP objects”

Object name	Description
`kernel-filter`	Opens the kernel-filter object for editing.
`deny-rule`	Specifies which traffic AA-SBC should block on the current IP interface.
`allow-rule`	Specifies which traffic AA-SBC should allow on the current IP interface.

kernel-filter

Purpose

Creates or edits kernel filters. Kernel filter rules allow you to deny traffic on an IP interface based on source IP address, source port number, and packet type.

Syntax

```
config cluster box integer interface ethX ip name kernel-filter
config cluster box integer interface ethX vlan integer ip name
   kernel-filter
config box interface ethX ip name kernel-filter
config box interface ethX vlan integer ip name kernel-filter
```

Properties

None

deny-rule

Purpose

Creates or edits the named kernel filter deny-rule configuration. A deny rule specifies the source IP address or subnet, source port number, and protocol associated with traffic to be blocked on the current IP interface.

Specify the rule name using up to 16 alphanumeric characters, enclosing blank spaces in quotation marks.

Syntax

```
config cluster box integer interface ethX ip name kernel-filter
   deny-rule name
config cluster box integer interface ethX vlan integer ip name
   kernel-filter deny-rule name
config box interface ethX ip name kernel-filter deny-rule name
config box interface ethX vlan integer ip name kernel-filter deny-rule
   name
```

Properties

Property name	Description
admin {enabled | disabled}	Sets the administrative state of this kernel filter deny rule. When enabled, network traffic is blocked using the configured IP address or subnet, port number, and packet type. When disabled, the deny rule is not in effect. **Example:** **`set admin enabled`** The default setting is **enabled**.
source-address/mask *ipAddress*/*mask*	Specifies the source IP address or subnet associated to filter (deny) on this IP interface. Specify the IP address and mask in CIDR format. **Example:** **`set source-address/mask 215.200.0.0/16`** The default source IP address is **0.0.0.0/0**.
source-port *portNumber*	Specifies the source port number associated with received packets to filter (deny) on this system interface. **Example:** **`set source-port 56`** The default source port number is **0** (deny all ports).
protocol {all | icmp| tcp | udp | vrrp}	Specifies the source protocol associated with received packets to filter (deny) on this system interface. **Example:** **`set protocol tcp`** The default protocol setting is **all** protocols.

allow-rule

Purpose

Creates or edits the named kernel filter allow-rule configuration. An allow rule specifies the source IP address or subnet, source port number, and protocol associated with traffic to be specifically allowed on the current IP interface. Typically the allow rule is used to override the denial of an subnet by allowing specific endpoints.

Specify the rule name using up to 16 alphanumeric characters, enclosing blank spaces in quotation marks.

Syntax

```
config cluster box integer interface ethX ip name kernel-filter
   allow-rule name
config cluster box integer interface ethX vlan integer ip name
   kernel-filter allow-rule name
config box interface ethX ip name kernel-filter allow-rule name
config box interface ethX vlan integer ip name kernel-filter
   allow-rule name
```

Properties

Property name	Description
admin {enabled | disabled}	Sets the administrative state of this kernel fitter allow rule. When enabled, network traffic is allowed using the configured IP address or subnet, port number, and packet type. When disabled, the allow rule is not in effect. **Example: `set admin enabled`** The default setting is **enabled**.
source-address/mask *ipAddress/mask*	Specifies the source subnet, but more typically IP address, to allow on this IP interface. Specify the IP address and mask in CIDR format. **Example: `set source-address/mask 215.200.40.8/32`** The default source IP address is **0.0.0.0/0**.

Property name	**Description**
source-port *portNumber*	Specifies the source port number associated with received packets to allow on this system interface. **Example:** **`set source-port 56`** The default source port number is **0** (allow all ports).
protocol {all | icmp| tcp | udp | vrrp}	Specifies the source protocol associated with received packets to allow on this system interface. **Example:** **`set protocol tcp`** The default protocol setting is **all** protocols.

38. Location service objects

Location service object description

AA-SBC uses the location service database (cache) to store SIP caller location (address-of-record) information. Since destinations can move to different locations, SIP INVITE messages are subject to lookup in the AA-SBC location service database. SIP enterprise servers also use this information to contact (call back) the originating SIP caller in a call session (SIP INVITE).

The location service database is updated using any one of the following methods:

- AA-SBC registration service
- static address-of-record (AOR) configuration records
- configured AA-SBC policies

A binding in the location cache can be in one of 13 states:

State	Description
requested	REGISTER request received.
trying	REGISTER forwarded and waiting.
responded	REGISTER response received.
aborted	REGISTER aborted from trying.
waiting	Waiting on server busy and will re-register in brief interval.
challenged	SIP 401/407 “Auth Required” response has been sent to the endpoint.
unauthenticated	Client did not responded to challenge in challenge-timeout period.
declined	REGISTER declined with proper code; AA-SBC continues to process subsequent REGISTERs.
rejected	All REGISTERs for this binding were rejected with proper code before session was created.
discarded	All REGISTERs for this binding were discarded silently before session was created.

State	Description
registered	This binding is valid and registered.
aged	This binding is aged but not deleted.
disconnected	The TCP/TLS connection for this binding is broken.
obsolete	This binding is obsolete; notification to upstream server is done.

See Chapter 56, “Registration service objects” for information on the service that provides address-of-record updates to the location service database.

Location service object summary

The following table lists and briefly describes the **location-service** objects. See the following chapter for other objects in the CLI hierarchy:

- Chapter 75, “Virtual system partition (VSP) objects”

Object name	Description
`location-service`	Opens the location-service configuration object for editing.
`settings`	Sets the storage location for address bindings and the cache polling interval.
`address-of-record`	Specifies one or more static SIP address-of-record entries that are entered in the AA-SBC location database.
`contact`	Creates static address-of-record bindings in the location service database.

location-service

Purpose

Opens the location-service configuration object from which you configure registration information, addresses of record, and database settings options.

Syntax

```
config vsp location-service
```

Properties

None

settings

Purpose

Sets the storage location for address bindings. These are the mappings of AORs to locations. Through this object you can also set the frequency of cache polling and limits on calling. When AA-SBC polls the cache, it also modifies the expiration time on entries and deletes those that have expired. Use the session configuration location-call-admission-control object to set admission control.

The location service database (the binding of AORs with locations) is a SQL database on AA-SBC. The database stores all learned location bindings; static records are not maintained in the location database as they are managed by configurations. However, all location record types are stored in the location cache (a binary tree-based table that contains all location bindings).

Several of the properties in this object can also be set in the address-of-record or registration-plan route objects. The address-of-record settings take precedence, however, as they are based on a more specific match. These route settings are next, and finally, these **settings** configurations serve as a default in the event of no other match.

Syntax

```
config vsp location-service settings
```

Properties

Property name	Description
persistent {false | true}	Stores location bindings into the SQL location database. Setting the property to **true** writes the bindings to the location database on disk. When set to **false**, location bindings are not saved in the disk; they are only saved in the cache. **Example:** set persistent false The default setting is **true**.
retention-mode {enabled | disabled}	Specifies whether the location cache remains unchanged if a failure event occurs (e.g., a server down event). If a failure occurs and this property is **enabled**, the system will not remove routes to that server from its location cache. If disabled, the system removes unreachable routes and relearns them when the server comes back on line. If the primary route is unavailable, the system forwards the call locally, using a dial plan with a different (available) server, another location cache binding, or DNS. **Example: set retention-mode disabled** The default setting is **enabled**.
alias-entry-timeout *seconds*	Specifies the number of seconds that an alias remains in the alias cache after a location cache registration has expired. A user can have many aliases. The system does not delete any alias for a user if the user has even one alias associated with a valid location cache entry. This value sets a timeout for the aliases, and is sent with a user registration. When the registration expires, the system then removes the alias when the timeout value has been reached. **Example: set alias-entry-timeout 90** The default setting is **60** seconds.

Property name	Description
cache-cleanup-interval *seconds*	Specifies the frequency with which the system purges the location cache of all bindings that are not in a registered state. When cleanup occurs, the system removes any binding that has aged (expired) or that has a TCP/TLS connection that is disconnected. It is a good idea to keep this interval fairly infrequent (the default is one day) because the location binding can contain state information that is helpful for troubleshooting. You can do an immediate cleanup of the cache at any time using the location action. **Example: set cache-cleanup-interval 30000** The default setting is **86400** seconds.
cache-poll-interval *seconds*	Sets the frequency with which the system polls the location cache. When the system polls the cache, it decrements the expiration time on each entry by the number of seconds of the poll interval. If an entry has an expiration time set to 0 as a result, the system removes that entry. **Example: set cache-poll-interval 30** The default setting is **60** seconds.
max-missing-registrations *integer*	Specifies the number of contiguous missing REGISTER requests the system will tolerate before deleting a binding from the location cache. This setting allows the system to accommodate inevitable network congestion. Use the verbose form of the **show location-binding** command to display the current count of missing REGISTERs. **Example: set max-missing-registrations 5** The default setting is **3** missing REGISTERs.

Property name	Description
challenge-timeout *seconds*	Specifies how long a binding can remain in a state of **challenged**. A binding enters the challenged state when the system sends a 401/407 “Auth Required” to an endpoint on behalf of a server. If the endpoint does not respond within the time configured with this property, the system changes the binding state to **unauthenticated**. **Example: set challenge-timeout 15** The default setting is **10** seconds.
max-cache-poll-duration *milliseconds*	Specifies the length of time, in milliseconds, that the system should work at purging the location cache. When the **cache-poll-interval** expires and the system polls the cache, it then begins deleting expired entries. It continues this processing until it has reached the number of entries specified with the **max-entries-per-poll** property. The system then checks the higher priority queues; if one of those queue thresholds has been reached, the system stops the purging process, regardless of whether this duration timer has expired. This prevents higher priority processing from being detained by the aging processing. (Set the urgent and priority queue levels with the virtual-threads object.) If the higher priority queues are sub-threshold, the resumes deleting until it reaches either the specified duration or the maximum number of entries again. **Example: set max-cache-poll-duration 1500** The default setting is **1000** milliseconds.
max-entries-per-poll *integer*	Specifies the number of entries that will be purged from the location cache before the system pauses to check the threshold status of the higher priority queues. See the **max-cache-poll-duration** description for more details. **Example: set max-entries-per-poll 150** The default setting is **100** entries.

Property name	Description
max-unregisters-per-poll *integer*	Sets the number of UNREGISTERs that can be sent as the result of a single poll. When a large number of bindings expire and the **unregistered-aged-binding** feature is enabled, if the system does not throttle the UNREGISTERs sent to the delegate using this feature, too much memory can be consumed. **Example: set max-unregisters-per-poll 750** The default setting is **1000** entries.
cache-binding-state {enabled | disabled}	Specifies whether to save intermediate (any state other than **registered**) binding states into the location cache. When **enabled**, all states are saved/changed according to other configuration parameters. When **disabled**, only bindings in the **registered** state are saved. Otherwise, the binding itself is deleted. **Example: set cache-binding-state disabled** The default setting is **enabled**.

Property name	**Description**
max-bindings-per-AOR *integer*	Specifies the maximum number of bindings allowed for each AOR. When an AOR reregisters, it may do so with a new binding, for example, because of a firewall configuration. When the maximum number of bindings is reached, the system overwrites the oldest binding in its tables if a new binding comes in. Keeping this value low controls the memory allocated to an AOR. The value must be high enough, however, to cover the number of locations your users might be registered from. If not, the system will not be able to do call forking. This value can be overwritten for specific matching AORs by the **max-bindings-per-AOR** property in the registration plan route object or the **max-bindings** property in the address-of-record object. Note that any change to this property requires a location **flush** action for the changes to take effect. **Example: set max-bindings-per-AOR 1** The default setting is **1** binding.
max-failed-bindings *integer*	Specifies the number of bindings allowed in the database of rejected bindings. When a binding is rejected, it is not installed in the location-binding database. Instead, it is written to the location-bindings-rejected database. This property sets the number of bindings allowed in that database. When the number is reached, the system continues writing to the database and deleting the older bindings necessary to stay within the threshold. **Example: set max-failed-bindings 1500** The default number of failed bindings is **1000**.

Property name	Description
binding-replacement {loose | tight | strict | custom *options*}	Specifies whether or not a subsequent REGISTER for a binding can replace the previous entry by setting match criteria for the new REGISTER request. If any of the parameters do not match, the system creates a new binding. After the system has delegated a REGISTER request, it will only process the response if it finds a valid binding for this response. This property sets a default behavior for a binding; it may be overwritten by a matching registration plan. When the following conditions are met for an option setting, the system replaces a binding if: • **loose**—the previous binding matches the public IP address. • **tight**—the previous binding matches the public IP address, port number, and transport protocol. • **strict**—the previous binding matches the public IP address, port number, and transport protocol, and the private IP address and port number. • **custom**—the previous binding matches the user-specified elements. Select from allowed options. **Example: set binding-replacement custom public-ip+** **public-port+call-ID** The default setting is **strict**.

Property name	Description
lookup-timeout *seconds*	Specifies the number of seconds a box within a cluster waits for the cluster master to download the location cache. When a system in a cluster receives a call, if the local box determines that it has no local cache entry for the AOR, it sends a request to the cluster master. If the cluster master does not deliver the entry to the local box within the time specified by this property, the local system dispatches the message to a worker thread. If the master still has not delivered the entry for the AOR by the time the thread has reached the point of processing, the system applies the action configured in the unregistered-sender-directive in the pre-session-config object. **Example: set lookup-timeout 2** The default setting is **5** seconds.
uri-case-sensitive {true | false}	Controls whether URI comparisons are case-sensitive or insensitive. When set to true, the AA-SBC does not change the case of characters in the URL (from upper to lower case). **Example: set uri-case-sensitive true** The default setting is **false**.

address-of-record

Purpose

Creates static address-of-record bindings in the AA-SBC location service database. Static bindings are those that you manually configure using this configuration object as opposed to address-of-record bindings received from a registration server, learned from a SIP client, or defined by configured policies. Many of the properties in this object can also be set in the registration-plan route object. These **address-of-record** settings take precedence, however, as they are based on a more specific match. Finally, the settings configuration serves as a default in the event of no other match.

These entries associate SIP recipients to specific domain names, as well as provide contact information for the SIP recipients. Address-of-record bindings map an incoming SIP or SIPS URI, such as sip:bob@company.com to one or more URIs that are more direct to that user, such as the extended sip:bob@marketing.company.com entry.

In addition, you can use the address-of-record to statically configure a default location and a variety of other parameters if it is not registered with AA-SBC. You should manually configure the location because usually SIP servers do not exchange REGISTER requests.

When specifying the SIP or SIPS (SIP secure) address-of-record, enter the word **sip** or **sips** followed by a colon (:), followed immediately (no space) by the uniform resource identifier (URI) string associated with the SIP user.

Admission control for an AOR

Admission control can be configured for all AORs (box-wide) in the settings object. You can set AOR-specific admission control through this object (**admission-control**, **emission-control**, **max-bandwidth**, and **max-concurrent-calls**). These settings override the default settings. You can also set session-specific admission control, which overrides any settings in location service, using the session configuration location-call-admission-control object.

Note that AOR admission control is not in effect if the AOR is registered via a configured SIP gateway. Only server CAC (admission and emission) and VSP CAC are available in those cases. If an AOR is registered using an IP gateway, it is only subject to the admission control that is present in the server configuration. Not only does the server configuration take precedence, but if not configured in the server, CAC doesn't apply (even if it is configured for the location-service).

An AOR must register with AA-SBC for the per-AOR admission control settings to take effect. Also, this feature only applies to established calls, which can result in exceeding the **max-concurrent-calls** threshold. For example, if **max-concurrent-calls** is set to 2, but 10 INVITEs come in simultaneously, all 10 calls could be setup. This is because at the time of each INVITE, there were zero calls fully established.

Syntax

```
config vsp location-service address-of-record {sip:|sips:}uri
```

Properties

Property name	Description
admin {enabled | disabled}	Sets the administrative state of this static address of record binding. **Example: set admin disabled** The default setting is **enabled**.
state {unregistered | trying | in-service | redirect | registered | out-of-service}	Sets state information that is saved with the AOR. Enter one of the following: • **unregistered**—the AOR is in an initial state. The system changes the state information when it receives information from the location service. • **trying**—the system temporarily accepts REGISTER requests for the AOR. Use this, for example, when consulting the location provider to verify registration privileges. • **in-service**—an initial acceptance of the AOR. When the system next receives a REGISTER request for the AOR, it changes the state to registered. This is the default. • **redirect**—the system redirects REGISTER requests intended for this AOR to the configured peer. Set the redirect destination using the **redirect-to-server** property. • **registered**—the system manages all REGISTER requests for the AOR. • **out-of-service**—the system does not provide registration services for the AOR; all REGISTER requests are declined. You can also change a location cache entry state using the location action. **Example: set state redirect** The default setting is **in-service**.

Property name	Description
access-control-level {strict | tight | loose}	Specifies the information an incoming call must match in an existing binding to be considered “known” by the system. Any call coming in to the system that does not meet the criteria is considered an unregistered sender. The system saves this requirement level setting, as part of the binding, in the location cache. When receiving a future call, the system performs a location cache lookup on the From URI to determine if it matches the necessary characteristics for the indicated requirement level. The registration is valid when: • **strict**— the previous binding matches the IP address, transport protocol, port number, and socket of the new binding. • **tight**— the previous binding has the same IP address as the new binding. • **loose**— the same AOR was registered previously. **Example: set registration-requirement-level strict** The default setting is **tight**.
do-not-disturb {enabled | disabled}	Sets the system to return a busy response to any call directed to this address of record. The phone registered to that AOR will respond according to its configuration (busy, voice mail, etc.). You can also use the set-do-not-disturb action to toggle an AOR “do not disturb” setting. **Example: set do-not-disturb enabled** The default setting is **disabled**.
call-forwarding {enabled | disabled}	Sets the system to forward any calls intended for this AOR to a specified URI. Specify the URI using the **call-forward-uri** property. You can also use the set-call-forwarding action to toggle an AOR call forwarding setting. **Example: set call-forwarding enabled** The default setting is **disabled**.

Property name	Description
call-forward-uri *uri*	Specifies the URI to which the system forwards calls when the **call-forwarding** property is set to **enabled**. **Example: set call-forward-uri sip:bob@company.com** There is no default setting.
redirect-to-server *ipAddress*/*Mask*	Specifies the server to which calls for this AOR should be redirected. This property is only applicable if the **state** property is set to **redirect**. Enter an address and mask or a host name. **Example: set redirect-to-server 10.10.20.5/24** There is no default setting.
redirect-to-transport {any | UDP | TCP | TLS}	Specifies the protocol to use when sending an AOR REGISTER request to the server it has been redirected to. **Example: set redirect-to-transport tls** The default setting is **UDP**.
redirect-to-port *portNumber*	Specifies the port to send to on the server identified in the **redirect-to-server** property. **Example: set redirect-to-port 5061** Enter a port number value between 1 and 65535. The default port is **5060**.

Property name	Description
trunk-port-type {disabled | per-AOR | per-binding | per-endpoint}	Sets the method the system uses to allocate listening ports. Select either: • **per-AOR**: allocates a single, unique listening port for all the bindings of an AOR. • **per-binding**: allocates a unique listening port for each binding, regardless of whether it is for the same AOR. • **per-endpoint**: allocates a unique listening port to all endpoints having the same IP address and port. This would be the case, for example, with devices behind a NAT. In all cases, the assigned port will also be used in the Contact header of the delegated REGISTER request and for all future transmissions targeted for this AOR. If the property is set to **disabled**, listening ports are not allocated to an AOR. The system allocates the port from the pool of available ports set with the media-ports object. **Example: set trunk-port-type per-AOR** The default setting is **disabled**.
admission-control {enabled | disabled}	Specifies whether the system considers AOR limitations when forwarding a call from this AOR. The system tracks the number of concurrent (both incoming and outgoing) active calls for this AOR. If this property is **enabled**, the system does not forward calls from this AOR if the limit has been reached and instead sends a “603 Declined” message. If **disabled**, the system does forward calls from the AOR. (Set the call limit with the **max-concurrent-calls** property.) See Admission control for an AOR for more information. **Example: set admission-control enabled** The default setting is **disabled**.

Property name	Description
emission-control {enabled | disabled}	Specifies whether the system considers AOR limitations when forwarding a call to this AOR. The system tracks the number of concurrent (both incoming and outgoing) active calls for this AOR. If this property is **enabled**, the system does not forward calls to this AOR if the limit, set with the **max-concurrent-calls** property, has been reached. Instead, the system sends one of the following messages and drops the call: • If there is one outbound server/UAC/UAS, the system sends a “486 Busy” message, indicating that the route was resolved but that the AOR was unavailable. • If there are multiple outbound server/UAC/UASs and all have reached the maximum concurrent calls threshold, the system sends a “486 Busy” message. • If there are multiple outbound server/UAC/UASs and at least one has not reached the maximum concurrent calls threshold, the return code is determined by the final server that the system attempted to reach. This could be, for example, “486 busy” or a “504 server timeout” if the last server was unresponsive and the transaction timed out. If **disabled**, the system continues to forward calls to this AOR. See Admission control for an AOR for more information. **Example: set emission-control enabled** The default setting is **disabled**.
max-bandwidth {*kbps* | unlimited}	Specifies the amount of bandwidth the system allocates to this AOR. When the system reaches the maximum bandwidth limit for a server, it rejects calls until bandwidth use drops below the maximum. **Example: set max-bandwidth 512** The default setting is **unlimited**.

Property name	Description
max-concurrent-calls *integer*	Specifies the maximum number of active incoming and outgoing calls allowed for this AOR at one time. When this number is reached, the system responds based on the configuration of the **action** property. For example, this value is used by the system when implementing admission and emission control. This would cause a decline or busy status until the value drops below the threshold. **Example: set max-concurrent-calls 4** The default setting is **2** calls.
action {accept | delegate | forward | redirect | tunnel | discard | block}	Specifies how the system processes any registration it receives that matches the AOR. See Configurable actions for registrations for a description of each action. **Example: set action redirect** The default type setting is **accept**.
response-code *code*	Sets the response code that the system sends to an endpoint when the **action** property is set to **accept** or **block**. (2xx response codes indicate success; change this value if the action is **block** and you have configured a **response-string**.) **Example: set response-code 201** The default response code is **200**.
response-string *string*	Sets the response string that the system sends to an endpoint when the **action** property is set to **accept** or **block**. **Example: set response-string “REGISTER was blocked”** There is no default response string.

Property name	Description
max-expiration *seconds*	Overwrites the client binding expiration time, as found in the client REGISTER request. The time you enter specifies the maximum time (in seconds) to elapse before a client REGISTER request becomes invalid and the registration information is removed from the location cache. If you enter 0, or as-requested, the client value remains. **Example: set max-expiration 4** The default setting is **3600** seconds.
min-expiration *seconds*	Overwrites the client's minimum expiration time, as found in the client REGISTER request. The time you enter specifies the minimum time (in seconds) to elapse before a client REGISTER request can become invalid and the registration information can be removed from the location cache. If you enter 0, or as-requested, the client value remains. **Example: set min-expiration 45** The default setting is **30** seconds.
max-bindings *integer*	Specifies the maximum number of bindings allowed for this AOR. When an AOR reregisters, it may do so with a new binding, for example, because of a firewall configuration. When the maximum number of bindings is reached, the system overwrites the oldest binding in its tables if a new binding comes in. **Example: set max-bindings-per-AOR 3** The default setting is **1** binding.

Property name	Description
edp [NAT] [TCP] [TLS]	*Secondary property.* Sets the connection type that the Expiration Discovery Process (EDP) is being used with, either NAT, TCP, and/or TLS. EDP is the process the system uses to detect a maximum of time in which system can reach an endpoint as indicated by the location binding, regardless of the expiration time set by the endpoint. With NAT, the selected expiration time keeps the NAT pinhole continually open for the endpoint—a firewall otherwise may age out a pinhole more quickly than the binding expiration. With TCP or TLS, the selected expiration time keeps the connection refreshed regularly and continually open for the endpoint. Otherwise, a TLS connection may age out because of TCP socket inactive timeout. **Example: set edp NAT** There is no default setting.
edp-expire-grow *seconds*	*Secondary property.* Specifies the number of seconds that the edp-expiration timer sent in the 200 OK message should increase or decrease by when the EDP process cycle receives a response to the system's OPTIONS message from an endpoint. When the EDP process is triggered, the edp-expiration timer starts, and the system changes a binding's state to WAITING. When the timer expires, the system sends an OPTIONS message to the endpoint and changes the state to PINGING. When the endpoint responds, the system changes the state to PINGED and the edp-expiration timer value is incremented by the value of **edp-expire-grow**. If the endpoint does not respond, the state is changed to TIMEOUT and the value of edp-expiration is decreased by the **edp-ping-timeout** value. (The value used in the previous cycle was the correct expiration time for the binding.) **Example: set edp-expire-grow 15** The default setting is **10** seconds.

Property name	Description
edp-ping-timeout *seconds*	*Secondary property.* Specifies the number of seconds added to the **min-client-expiration** value to set the EDP expiration time that is sent in the 200 OK message. If that new value (the sum of **min-client-expiration** and **edp-ping-timeout**) is less than the original expiration, the system triggers the EDP process. **Example: set edp-ping-timeout 45** The default setting is **30** seconds.
uac-preferred-contact {auto | public | private}	*Secondary property.* Determines where the Host portion of the INVITE Request URI or To header is derived from. Select either: • **auto**—the Host portion is determined automatically. If a REGISTER is received from a SIP proxy, then the host is set to private. • **public**—if the caller is behind a firewall, the Host portion is set to the public IP address of the firewall (the NAT address). • **private**—if the caller is behind a firewall, the Host portion is set to the private IP address of the UAC. **Example: set uac-preferred-contact private** The default setting is **auto**.
routing-tag *string*	*Secondary property.* Controls which outbound interface SIP traffic uses. The routing-tag indicates the interface on the server where a SIP message with a matching routing-tag would be forwarded. The SIP message derives its routing-tag from the session config or IP interface classification-tag, depending on the configuration scenario. This property sets the initial routing tag for an AOR. If there is a policy match that applies to the AOR, and that configuration sets a routing tag (with the routing-settings ingress- and egress-classification-tag), the policy setting takes precedence. **Example: set routing-tag tag1** There is no default setting.

Property name	**Description**
calling-group *callingGroupReference*	*Secondary property.* Associates the AOR with a configured calling-group group. Therefore, you can segregate routing arbitration, call routing, policy, and normalization based on the user group. **Example: set calling-group “vsp calling-groups group east2”** There is no default setting.
registration-throttling {enabled | disabled}	Sets whether the system responds locally to a registration request. When **enabled**, if the system has an existing binding for the contact in the REGISTER request and that binding has not expired on the peer, the system forgoes registration. When throttled, the system also does not perform proxy authorization. If **disabled**, the system forwards all registration requests to the delegate server. **Example: set registration-throttling disabled** The default setting is **enabled**.

Property name	Description
binding-replacement {loose | tight | strict | custom}	Determines whether or not a subsequent REGISTER for a binding can replace the previous entry by setting match criteria for the new REGISTER request. If any of the parameters do not match, the system creates a new binding. After the system has delegated a REGISTER request, it will only process the response if it finds a valid binding for this response. When the following conditions are met for an option setting, the system replaces a binding if: • **loose**—the previous binding matches the public IP address. • **tight**— the previous binding matches the public IP address, port number, and transport protocol. • **strict**—the previous binding matches the public IP address, port number, and transport protocol, and the private IP address and port number. • **custom**—the previous binding matches the user-specified elements. **Example: set binding-replacement custom public-ip+public-port+ call-ID** The default setting is **strict**.
authentication-interval *seconds*	Specifies how frequently the system re-authenticates the AOR. Once an AOR has registered, the system throttles future registrations, acting as a proxy for the registrar. This property sets how frequently the system lets a REGISTER through to re-authenticate the AOR. **Example: set authentication-interval 30000** The default setting is **86400** seconds.

contact

Purpose

Creates static address-of-record bindings (caller contact information) in the AA-SBC location service database. Static bindings are those that you manually configure using the address-of-record configuration object as opposed to address-of-record bindings received from a registration server, learned from a SIP client, or defined by configured policies. Typically an AOR would have at least two contacts (bindings), with each having a different host address and device type.

Enter a display name, usually a full name. This name is used by AA-SBC to uniquely identify the contact object, not to search for the appropriate binding. Note that if the contact begins with “sips:”, the **transport** property (protocol) must be TLS.

Syntax

```
config vsp location-service address-of-record {sip:|sips:}uri contact
   displayName
```

Properties

Property name	Description
user *string*	Specifies a user name for this contact, for example, a login name. **Example: set user jdoe** There is no default setting.
host *ipAddress*	*Required*. Specifies the location the system should use to reach this contact. Enter a host name or IP address. **Example: set host 10.0.10.10** There is no default setting.

Property name	Description
param *string*	Appends the specified user parameter and value to the contact SIP address. For example, the example below would result in a URI that looked similar to: <sip:jane@fun.com;user=spot> **Example: set param spot** There is no default setting.
port *portNumber*	Specifies the contact port on the system used to reach this user. **Example: set port 5061** Enter a port number between 1 and 65535. The default port is **5060**.
state *state*	Sets state information for the user. **Example: set state declined** The default setting is **registered**.
external-contact-ip *ipAddress*	Specifies the public IP address that the contact uses when behind a virtual firewall.) **Example: set external-contact-ip 192.168.100.10** There is no default setting.
transport {any | UDP | TCP | TLS}	Specifies the protocol used with this contact. Note that if the contact display name begins with “sips,” the protocol must be TLS. **Example: set transport tls** The default setting is **UDP**.
is-secure {true | false}	Indicates whether communication to the AOR at this location is secure or not. If set to **true**, SIP signaling and media exchanges must use a secure protocol, such as TLS. **Example: set is-secure true** The default setting is **false**.
device-type *type*	Specifies the type of device that this binding is representing. Use the question mark at the command line for a current list of supported device types. **Example: set device-type windows-messenger** The default setting is **sipura**.

Property name	Description
device-user-agent *string*	Adds an informational field describing the device type. Use quotation marks if entering multiple words with spaces. **Example: set device-user-agent "Version 7.0"** There is no default setting.
local-port *portNumber*	Specifies which port on the system is used to communicate with the device. **Example: set local-port 5061** Enter a port number value between 1 and 65535. The default port is **5060**.
proxy-trunk-port *portNumber*	Specifies the port to use if the system proxies or delegates the AOR to another server or gateway. **Example: set proxy-trunk-port 5061** Enter a port number value between 1 and 65535. The default port is **5060**.

39. Master services objects

Master services description

The master services objects allow you to enable services for directory, accounting, authentication, database, registration, media failover, and cluster services. Each master service can run on one box in the cluster. The **host-box** property within each master service object defines the primary box for that service. You can also configure backup boxes in the event of primary failure by re-executing the **host-box** configuration.

In the example below, the first host box listed in the configuration serves as the primary host for the directory service. Subsequent host boxes (2 and 3) serve as backup.

```
NNOS-E> config master-services
config master-services> config directory
config directory> set host-box cluster box 1
config directory> set host-box cluster box 2
config directory> set host-box cluster box 3

config directory> show
 master-services
  directory
   admin enabled
   host-box[1] cluster\box 1
   host-box[2] cluster\box 2
   host-box[3] cluster\box 3
```

The first **host-box** property identifies which box runs the service. If that box does not perform, the other configured host boxes will perform in succession and attempt to boot the service.

Master services in VRRP configurations

Master services that are running in vrrp configurations can use the **group** property as an additional backup mechanism. The group property is an option to link the VRRP configuration with other services on the box, in this case the master services. If one interface of a VRRP pair is down, the group with which they are associated is considered down. If a service is associated with that group, the box hosting the downed VRRP pair stops the service, and the backup box then restarts it. (A vinterface can have more than one Ethernet interface for a given box. AA-SBC does not bring the service down until all configured Ethernet interfaces have been established as “unavailable.”)

This feature is illustrated in the following sample output. (Note that display of properties unrelated to this feature have been removed for clarity.) If either eth1 *and* eth4 or eth2 on box 1 lose link, AA-SBC considers VRRP group 1 down. This causes both vinterface vx1 and vx2 on box 1 to go to their configured backups, resulting in box 2 becoming master for both of these VRRP interfaces.

In the master services configuration, the directory service is in group 1 but accounting is not associated with a VRRP group. This configuration results in AA-SBC causing box 1 to stop the directory service and box 2, as backup, to restart it. The accounting service remains unchanged as it was not associated with the VRRP configuration.

```
config vrrp> show -v
cluster
  vrrp
    admin enabled
    vinterface vx1
      admin enabled
      group 1
      host-interface[1] cluster\box 1\interface eth1
      host-interface[2] cluster\box 1\interface eth4
      host-interface[3] cluster\box 2\interface eth1
    vinterface vx2
      admin enabled
      group 1
      host-interface[1] cluster\box 1\interface eth2
      host-interface[2] cluster\box 2\interface eth2
```

```
config master-services> show -v
directory
  admin enabled
  host-box[1] cluster\box 1
  host-box[2] cluster\box 2
  group 1
```

```
accounting
  admin enabled
  host-box[1] cluster\box 2
  host-box[2] cluster\box 1
  group 0
```

Master services object summary

The following table lists and briefly describes the **master-services** objects.

Object name	Description
master-services	Opens the master services configuration object. for editing.
cluster-master	Specifies the box that maintains the master configuration.
directory	Enables and disables master directory services.
settings	Configures memory allowances for certain master service processes.
accounting	Enables and disables master accounting services.
settings	Configures memory allowances for certain master service processes.
authentication	Enables and disables master authentication services.
database	Enables and disables master database services.
registration	Enables and disables registration database master services.
server-load	Enables load balancing to support arbiter least-load routing algorithm.
call-failover	Enables and disables failover for the media process.
load-balancing	Enables and disables load balancing with head-end and backing interfaces.
file-mirror	Sets all participating AA-SBC devices to share particular files.
route-server	Sets the least-cost routing master service.

Object name	Description
sampling	Opens the mechanism for setting the interval at which AA-SBC samples operational aspects of the system for, either, display in the AA-SBC Management System or for sending to an IBM Tivoli server.
tivoli	Configures the IBM Tivoli server as a target for sampling data.
database	Configures the local database as a target for sampling data.
status	Sets the status providers that report to the parent target.
provider	Configures the providers to report to the parent target.
jtapi	
settings	Configures memory allowances for certain master service processes.
available-memory	Enables a sample interval for the AA-SBC to check the available memory.

master-services

Purpose

Opens the master services configuration object.

Syntax

```
config master-services
```

Properties

Property name	Description
advertisement-interval *seconds*	*Secondary property. S*ets the interval at which the master services broadcast their locations to boxes in the cluster. This setting applies to all master services. **Example: set advertisement-interval 90** The default setting is **60** seconds.
boot-interval *seconds*	*Secondary property.* For debugging only. Sets the wait time for restart of master services at system boot in a non-clustered network. In a clustered network, master-services start once the clustering is established (usually less than 30 seconds). **Example: set boot-interval 60** The default setting is **30** seconds..

cluster-master

Purpose

Configures the box that maintains the master configuration for the cluster. It pushes out configuration changes to other boxes in the cluster. If a different box becomes cluster-master, it would then start sending out its configuration to the other boxes.

Syntax

```
config master-services cluster-master
```

Properties

Property name	Description
admin {enabled | disabled}	Enables and disables the configuration that selects a master directory service. Enabling the directory service provides a link to the system gateways. **Example: set admin enabled** The default setting is **enabled**.
host-box *boxReference*	Specifies the box that acts as cluster master. You must select a box to serve as the cluster master using this property. See Master services description for more information. **Example: set host-box cluster box 1** There is no default setting.
group *groupID*	Associates the cluster master with a VRRP group. Enter the number of a previously configured vrrp group. See Master services in VRRP configurations for a complete explanation. **Example: set group 1** Enter a value for the VRRP group. The default setting is 0 (no grouping association).
preempt {true | false}	*Secondary property.* Specifies whether the master service should retake the mastership if it has gone down and then returned to operation. If set to true, the master resumes its position. If set to false, the backup service retains master control. **Example: set preempt true** The default setting is **false**.
takeover-timer-value *milliseconds*	*Secondary property.* Specifies the number of milliseconds that the master service stays in “awaiting takeover” mode at boot time. When a box boots, each hosted master service waits for this period of time to determine if any existing boxes in the cluster are already running that service before assuming mastership. **Example: set takeover-timer-value 2000** The default setting is **1000** milliseconds.

directory

Purpose

Opens the AA-SBC directory services object on the master. Directory services is the function that allows communication between AA-SBC and gateway services such as Active Directory, LCS, Sametime, etc. If this object is disabled, you can still configure the enterprise gateway services (through the directory object), but they do not become active until you enable this master service.

Note that if you have enabled the **local-directory-based-user-services** property for VSP settings, you must configure directory services on at least one box in the cluster.

See Chapter 27, “Enterprise objects” for information on enabling the directory service.

Syntax

```
config master-services directory
```

Properties

Property name	Description
admin {enabled | disabled}	Enables and disables the system directory service. Enabling the directory service provides a link to the system gateways. **Example: set admin enabled** The default setting is **enabled**.
host-box *boxReference*	Specifies the master box on which directory services run and, optionally, backup boxes. See Master services description for more information. If there is no host box specified, no directory services can run. **Example: set host-box cluster box 2** There is no default setting.
group *groupID*	Associates the directory service with a VRRP group. Enter the number of a previously configured vrrp group. See Master services in VRRP configurations for a complete explanation. **Example: set group 1** Enter a value for the VRRP group. The default setting is **0** (no grouping association).
preempt {true | false}	*Secondary property.* Specifies whether the master service should retake the mastership if it has gone down and then returned to operation. If set to true, the master resumes its position. If set to false, the backup service retains master control. **Example: set preempt true** The default setting is **false**.
takeover-timer-value *milliseconds*	*Secondary property.* Specifies the number of milliseconds that the master service stays in “awaiting takeover” mode at boot time. When a box boots, each hosted master service waits for this period of time to determine if any existing boxes in the cluster are already running that service before assuming mastership. **Example: set takeover-timer-value 2000** The default setting is **1000** milliseconds.

settings

Purpose

Secondary object. Configures total and minimum memory allowance for the directory, accounting, and/or database processes. Use these properties to fine-tine the default settings for applications that require additional memory.

Syntax

```
config master-services directory settings
config master-services accounting settings
config master-services database settings
config master-services jtapi settings
```

Properties

Property name	Description
heap-max *megabytes*	Specifies the total amount of memory that the system can allocate to the directory, accounting, or database processes. **Example: set heap-max 256** The default setting is **128** MB. Enter a value between 16 and 2,048.
heap-min *megabytes*	Specifies the minimum amount of memory that the system can allocate to the directory, accounting, or database processes. **Example: set heap-min 96** The default setting is **32** MB. Enter a value between 16 and 2,048.
argument *argument*	For Technical Support use only.

Property name	Description
thread-checker {true | false}	Configures the system to check for blocked threads in current processes. If **true**, the system monitors for blocked threads at a regular interval. If a process is repeatedly blocked, AA-SBC forces a restart of the process. It then writes a message to the event log, indicating the process name and ID, and recording a stack trace. This property is for Technical Support use only. **Example: set thread-checker true** The default setting is **false**.
association-min-lifetime	This is a secondary property. Controls the amount of time in seconds that “association” information (data for a given to-from URI pair) is preserved. **Example: set association-min-lifetime 25000** The minimum configuration setting for this property is 0. The maximum is 360000. The default setting is **300**.
max-proxy-transactions-per-session	*Secondary property*. Sets the maximum number of concurrent proxy transactions that a session can have. The minimum valid value is 1 and the maximum valid value is 65535. **Example: set max-proxy-transactions-per-session 30** The default setting is **20**.

accounting

Purpose

Opens the AA-SBC accounting services object running on the master. Accounting services is the object that enables or disables all accounting functions on AA-SBC, such as RADIUS and Diameter accounting services, system logging (syslog), the accounting database, and the accounting file system. If this object is disabled, you can still configure the accounting services, but they do not become active until you enable this master service. This setting overrides the setting of each individual accounting function.

See Chapter 6, “Accounting objects” for information on enabling and configuring accounting services. See settings for information on memory allowance settings.

Syntax

```
config master-services accounting
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables accounting services on the system. **Example: set admin enabled** The default setting is **enabled**.
host-box *boxReference*	Specifies the master box on which accounting services run and, optionally, backup boxes. See Master services description for more information. If there is no host box specified, no accounting services can run. **Example: set host-box cluster box 1** There is no default setting.
group *groupID*	Associates the accounting service with a VRRP group. Enter the number of a previously configured vrrp group. See Master services in VRRP configurations for a complete explanation. **Example: set group 1** Enter a value for the VRRP group. The default setting is **0** (no grouping association).
preempt {true | false}	*Secondary property.* Specifies whether the master service should retake the mastership if it has gone down and then returned to operation. If set to true, the master resumes its position. If set to false, the backup service retains master control. **Example: set preempt true** The default setting is **false**.
takeover-timer-value *milliseconds*	*Secondary property.* Specifies the number of milliseconds that the master service stays in “awaiting takeover” mode at boot time. When a box boots, each hosted master service waits for this period of time to determine if any existing boxes in the cluster are already running that service before assuming mastership. **Example: set takeover-timer-value 2000** The default setting is **1000** milliseconds.

authentication

Purpose

Opens the AA-SBC authentication services object running on the master. This object enables or disables all authentication functions on AA-SBC, such as RADIUS and Diameter authentication services, and local user profiles. If this object is disabled, you can still configure the authentication services, but they do not become active until you enable this master service. This setting overrides the setting of each individual authentication function.

See Chapter 54, “RADIUS-group objects” and Chapter 22, “Diameter client and server objects” for information on enabling and configuring authentication services.

Syntax

```
config master-services authentication
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables authentication services on the system. **Example: set admin enabled** The default setting is **enabled**.
host-box *boxReference*	Specifies the master box on which authentication services run and, optionally, backup boxes. See Master services description for more information. If there is no host box specified, no authentication services can run. **Example: set host-box cluster box 1** There is no default setting.
group *groupID*	Associates the authentication service with a VRRP group. Enter the number of a previously configured vrrp group. See Master services in VRRP configurations for a complete explanation. **Example: set group 1** Enter a value for the VRRP group. The default setting is **0** (no grouping association).
preempt {true | false}	*Secondary property.* Specifies whether the master service should retake the mastership if it has gone down and then returned to operation. If set to true, the master resumes its position. If set to false, the backup service retains master control. **Example: set preempt true** The default setting is **false**.
takeover-timer-value *milliseconds*	*Secondary property.* Specifies the number of milliseconds that the master service stays in “awaiting takeover” mode at boot time. When a box boots, each hosted master service waits for this period of time to determine if any existing boxes in the cluster are already running that service before assuming mastership. **Example: set takeover-timer-value 2000** The default setting is **1000** milliseconds.

database

Purpose

Opens the AA-SBC database object running on the master. This is the system database that contains the traffic data that resulted from tracing all packets in and out of AA-SBC. The system writes the header information from all TCP, UDP, and IP packets to the database, as well as all fields in the SIP header. This database is used by the system for DOS analysis, and in the AA-SBC Management System to display call details and call sequence.

Note: The database-write property must be enabled in the vsp object for AA-SBC to write data to the database.

From the database object you can also set up operations for cleaning the database. The maintenance operations are based on the SQL VACUUM command, which reclaims storage occupied by deleted entries. Maintenance purges the database of old entries at regularly scheduled intervals. An entry is considered “old,” and is therefore purged, if it is older than the day limit for the history table. Use the database object database subobject to set the number of days worth of entries to keep, based on table type.

If you set the maintenance period to zero, you disable the function. You can execute an immediate database purge using the top-level database action. Note that you may still see indication of entries in the session table if the corresponding history table entries have not yet aged out.

See settings for information on memory allowance settings.

Syntax

```
config master-services database
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables the system database on the system. If the database is disabled, the system cannot perform DOS analysis or record call details. **Example: set admin enabled** The default setting is **enabled**.
host-box *boxReference*	Specifies the master box on which database services run and, optionally, backup boxes. See Master services description for more information. If there is no host box specified, no database services can run. **Example: set host-box cluster box 1** There is no default setting.
group *groupID*	Associates the database service with a VRRP group. Enter the number of a previously configured vrrp group. See Master services in VRRP configurations for a complete explanation. **Example: set group 1** Enter a value for the VRRP group. The default setting is **0** (no grouping association).

Property name	Description
maintenance {period *hours* | time-of-day *hour*:*minutes* | disabled}	Sets the time for or frequency of database purging. If a daily purge is not appropriate, use the **period** property to set the number of hours between executions. However, it is advisable to run maintenance at least every 24 hours. Enter either: • **period**—the regularity with which maintenance should occur. Enter a number of hours. Entering 0 disables the maintenance function; this value should only be used for troubleshooting purposes. • **time-of-day**—the time at which the maintenance should occur. Enter a time in 24-hour format (for example, enter 17:00 for 5:00 p.m.). The system uses local time. • **disabled**—turns off the database purging feature. Be aware that if you disable maintenance, the database files can get quite large. Use this option as a debugging tool and then re-enable maintenance. **Example: set maintenance time-of-day 0:00** The default setting is **time-of-day 03:00**.
media {enabled | disabled}	Specifies whether the system writes media information to the database (to the media message table). The system does not create entries in the media message table if this property is not enabled. If it is enabled, use the show database-tables command to display the number of messages recorded in the table. Also, this property must be enabled to use the Play and Call-out links of the AA-SBC Call Logs tab. If disabled, there is some performance increase due to fewer writes to the database. **Example: set media enabled** The default setting is **disabled**.

Property name	Description
preempt {true | false}	*Secondary property.* Specifies whether the master service should retake the mastership if it has gone down and then returned to operation. If set to true, the master resumes its position. If set to false, the backup service retains master control. **Example: set preempt true** The default setting is **false**.
takeover-timer-value *milliseconds*	*Secondary property.* Specifies the number of milliseconds that the master service stays in “awaiting takeover” mode at boot time. When a box boots, each hosted master service waits for this period of time to determine if any existing boxes in the cluster are already running that service before assuming mastership. **Example: set takeover-timer-value 2000** The default setting is **1000** millisecond.
database-threads-max *threads*	*Secondary property.* Sets the number of threads dedicated to database operation and maintenance. The minimum number of threads is two—one for writing to the database, and a separate thread for database maintenance (e.g., purging old records). By increasing the number of threads, you can improve database write performance, allowing multiple threads to write to the database simultaneously. **Example: set database-threads-max 4** Enter a value between 2 and 4. The default setting is **2** threads.
sip-cache-size *entries*	*Secondary property.* Sets the number of entries allowed in the cache for the SIP database. When the system reaches the configured limit, it begins dropping entries, oldest first. **Example: set sip-cache-size 3000** Enter a value between 100 and 10,000. The default setting is **1000** entries.

Property name	Description
performance {call-rate | call-details}	*Secondary property.* Sets the point of optimization for calls: • **call-rate**—the database cache works harder to increase call-rate, at the expense of call details. (This effects the AA-SBC Management System display of such things as call diagrams and the SIP Messages viewer.) • **call-details**—causes the system to accurately write out individual records until the database reaches the configured queue-depth. (When reached, the system begins caching the records.) **Example: set performance call-rate** The default setting is **call-details**.
dos-tcp-connect-multiplier *integer*	*Secondary property.* Sets the number of “hits” that the system should count for each connection (as opposed to data packets). With each data packet that matches a pattern, the system counts the match as an event; when the event count reaches a set threshold, it creates a DOS rule. Each connection, because it is more compute intensive, can count as more than one event. Set this multiplier to the number of events the system should count for each TCP connection. **Example: set dos-tcp-connect-multiplier 3** The default setting is **5**.
dos-tls-connect-multiplier *integer*	*Secondary property.* Sets the number of “hits” that the system should count for each connection (as opposed to data packets). With each data packet that matches a pattern, the system counts the match as an event; when the event count reaches a set threshold, it creates a DOS rule. Each connection, because it is more compute intensive, can count as more than one event. Set this multiplier to the number of events the system should count for each TLS connection. **Example: set dos-tls-connect-multiplier 12** The default setting is **10**.

Property name	Description
sip-registers {enabled | disabled | cached}	*Secondary property.* Specifies whether to cache SIP REGISTER messages in the SIP message database. By disabling these messages, you can experience some performance increase. When **enabled**, the system writes them to the database in real-time. When set to **cached**, the system writes them to the database once every 5 minutes. Note that a more efficient way to disable writing of SIP REGISTER messages is to set the **message-logging** property of the log-alert object to **no-registers**. **Example: set sip-registers cached** The default setting is **enabled**.
max-queue-depth *messages*	*Secondary property.* Sets the maximum number of write requests allowed in the database queue. When this value is reached, the system begins dropping requests. **Example: set max-queue-depth 4500** The default setting is **4000**.
caching-threshold *messages*	*Secondary property.* Sets the point in the queue at which the system begins using cached entries instead of single writes to the database. When the queue depth reaches the threshold set with this property, the system uses the resource-efficient method of scanning the cache and writing all entries for a particular row at once. **Example: set caching-threshold 4000** The default setting is **3500**.
write-mode {insert | copy}	*Secondary property.* Controls the way the system writes records to the database. Typically, the system inserts records into the database. When set to **copy**, the system activates an experimental database optimization method. Use this option only if instructed to do so by Technical Support. **Example: set write-mode copy** The default setting is **insert**.

registration

Purpose

Configures the master service for the registration process. In a cluster, the registration database only runs on the specified master and the selected backups. The **host-box** property establishes the master and selective mirroring. The first box listed is the master, while subsequent boxes have mirrored databases. Boxes not configured via the **host-box** property do not run the registration database. Instead, they use a local location cache. This object must be enabled for load-balancing of SIP processing (across backing interfaces configured via the sip object) to function correctly.

Syntax

```
config master-services registration
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables registration services on the system. If **disabled**, the system cannot perform intracluster registration lookups. Also, you lose the persistence for registrations, which is what allows the system to perform rollover or failover operations. **Example: set admin enabled** The default setting is **enabled**.
host-box *boxReference*	Specifies the master box on which the registration service runs and, optionally, backup boxes. See Master services description for more information. If there is no host box specified, the registration service is not available to the system. **Example: set host-box cluster box 1** There is no default setting.

Property name	Description
group *groupID*	Associates the registration service with a VRRP group. Enter the number of a previously configured vrrp group. See Master services in VRRP configurations for a complete explanation. **Example: set group 1** Enter a value for the VRRP group. The default setting is **0** (no grouping association).
mirror-all-entries {enabled | disabled}	Specifies whether to mirror location cache entries to all boxes in the cluster. When **enabled**, all entries are mirrored to all boxes; when **disabled**, they are not. Mirroring helps failover recovery. In addition, you must set this property to enabled when the route object **alter-contact** property is set to **trunk-port-per-aor** for port forwarding. **Example: set mirror-all-entries disabled** The default setting is **enabled**.
mirror-location-cache {enabled | disabled}	Specifies whether AA-SBC mirrors the location cache to the other systems in the cluster. Disable this setting if the cluster is to handle up to one million SIP REGISTER requests. When **enabled,** the registration process mirrors the location cache entries from the cluster to each SIP process, giving a slight performance improvement while limiting the total cluster to 250,000 users **Example: set mirror-location-cache disabled** The default setting is **enabled**.
cache-poll-interval *seconds*	Configures a timer that scans registration data and purges stale bindings and/or cache entries. This property controls how often the task repeats. To turn this feature off, set the **cache-poll-interval** to 0. **Example: set cache-poll-interval 43200** The default setting is **86400** seconds.
max-poll-duration *milliseconds*	Sets how long the database remains unlocked between polling of *max-entries.* **Example: set max-poll-duration 1200** The default setting is **86400** seconds.

Master services objects

Property name	Description
max-entries-per-poll *entries*	Sets the number of registration data entries that are scanned at a time. This is a performance optimization setting, which helps in preventing the database from being locked for excessive periods of time. **Example: set max-entries-per-poll 1200** The default setting is **100** seconds.
preempt {true | false}	*Secondary property.* Specifies whether the master service should retake the mastership if it has gone down and then returned to operation. If set to true, the master resumes its position. If set to false, the backup service retains master control. **Example: set preempt true** The default setting is **false**.
takeover-timer-value *milliseconds*	*Secondary property.* Specifies the number of milliseconds that the master service stays in “awaiting takeover” mode at boot time. When a box boots, each hosted master service waits for this period of time to determine if any existing boxes in the cluster are already running that service before assuming mastership. **Example: set takeover-timer-value 2000** The default setting is **100** milliseconds.
force-regdb-lookup {enabled | disabled}	*Secondary property.* Sets whether AA-SBC does a registration database lookup on every request. When **enabled**, the system does the lookup, ensuring that a cache entry always has cluster-wideup-to-date information. Use this when the bindings of an AOR are distributed on different backing boxes to ensure that the registration database has the complete list of bindings and the system can complete call forking in a failover scenario. **Example: set force-regdb-lookup enabled** The default setting is **disabled**.

Property name	Description
ignore-from-tag {enabled | disabled}	When enabled, the AA-SBC uses the call ID only to associate the registration with a session. When disabled, the AA-SBC uses both the call ID and the From tag to associate the registration to a session. **Example: set ignore-from-tag disabled** The default setting is **enabled**.
fatal-error-code	Enter the response code for a fatal error. The following are fatal errors: • No registration-plan • No proxy-contact allowed • DNS lookup failure **Example: set fatal-error-code 500** The minimum configuration setting for this parameter is 400. The maximum is 699. The default setting is **403**.
fatal-error-string	Enter the text to be used in the fatal error. **Example**: **set fatal-error-string error1** The default setting is **Forbidden**.

Property name	**Description**
temporary-failure-code	Enter the response code for the temporary failure. The following are temporary failures: • Out of memory conditions • Next hop host is down • Unable to allocate trunk ports **Example**: **set temporary-failure-code 300** The minimum configuration setting for this parameter is 400. The maximum is 699. The default setting is **403**.
temporary-failure-string	Enter the text to be used in a temporary error. **Example**: **set temporary-failure-string failure1** The default setting is **Forbidden**.

server-load

Purpose

Configures AA-SBC to calculate server load. This object must be enabled if your dial plan arbiter settings use **least-load** as the routing algorithm option. (The arbiter **rules** property sets the criteria by which AA-SBC selects the server to which it forwards calls.)

Syntax

```
config master-services server-load
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables server load calculation on the system. If **disabled**, the dial plan arbiter cannot use the least-load routing arbitration rule. **Example: set admin enabled** The default setting is **enabled**.
host-box *boxReference*	Specifies the master box on which the server load calculation runs and, optionally, backup boxes. See Master services description for more information. If there is no host box specified, the calculation service is not available to the system. Box 1 is the default box. Your first entry overwrites that default. Subsequent entries are added to the list as backup boxes. **Example: set host-box cluster box 2** The default setting is box **1**.
group *groupID*	Associates the calculation service with a VRRP group. Enter the number of a previously configured vrrp group. See Master services in VRRP configurations for a complete explanation. **Example: set group 1** Enter a value for the VRRP group. The default setting is **0** (no grouping association).

Property name	Description
preempt {true | false}	*Secondary property.* Specifies whether the master service should retake the mastership if it has gone down and then returned to operation. If set to true, the master resumes its position. If set to false, the backup service retains master control. **Example: set preempt true** The default setting is **false**.
takeover-timer-value *milliseconds*	*Secondary property.* Specifies the number of milliseconds that the master service stays in “awaiting takeover” mode at boot time. When a box boots, each hosted master service waits for this period of time to determine if any existing boxes in the cluster are already running that service before assuming mastership. **Example: set takeover-timer-value 2000** The default setting is **1000** milliseconds.
update-timer *milliseconds*	*Secondary property.* Sets how often the server-load master updates the other boxes in the cluster. **Example: set update-timer 3500** The default setting is **5000** milliseconds.

call-failover

Purpose

Configures failover for the media and signaling streams. As a master service, the configured host box distributes copies of the media and kernel rules to all backup boxes in a cluster. AA-SBC uses the database on the host box, but enabling this master service ensures that there is an active copy of the database on another box in the cluster in the event of a failure.

Syntax

```
config master-services call-failover
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables failover for media services on the system. If **disabled**, the dial plan arbiter cannot use the least-load routing arbitration rule. **Example: set admin enabled** The default setting is **enabled**.
host-box *boxReference*	Specifies the master box on which the media services database runs and, optionally, backup boxes. See Master services description for more information. If there is no host box specified, the calculation service is not available to the system. Box 1 is the default box. Your first entry overwrites that default. Subsequent entries are added to the list as backup boxes. **Example: set host-box cluster box 2** The default setting is box **1**.
group *groupID*	Associates the media service with a VRRP group. Enter the number of a previously configured vrrp group. See Master services in VRRP configurations for a complete explanation. **Example: set group 1** Enter a value for the VRRP group. The default setting is **0** (no grouping association).

Property name	Description
preempt {true | false}	*Secondary property.* Specifies whether the master service should retake the mastership if it has gone down and then returned to operation. If set to true, the master resumes its position. If set to false, the backup service retains master control. **Example: set preempt true** The default setting is **false**.
takeover-timer-value *milliseconds*	*Secondary property.* Specifies the number of milliseconds that the master service stays in “awaiting takeover” mode at boot time. When a box boots, each hosted master service waits for this period of time to determine if any existing boxes in the cluster are already running that service before assuming mastership. **Example: set takeover-timer-value 2000** The default setting is **1000** milliseconds.
server-load {enabled | disabled}	When enabled, the AA-SBC calculates the server load and distributes traffic counters around the cluster. Based on these distribution counts, each AA-SBC in a cluster knows the fail-over status. **Example: set server-load enabled** `The default setting is` **`disabled.`**

load-balancing

Purpose

Configures boxes to host the load-balancing master service. These boxes are responsible for keeping the rule database up-to-date. They do not need to be the same boxes as the ones that host the head-end interfaces, although it is common to do so. (You can, for example, configure boxes in the cluster that only serve as host boxes, without any head-end interfaces or backing interfaces.)

Syntax

```
config master-services load-balancing
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables maintenance of the rule database for the purposes of load balancing. If **disabled**, the sip load-balancing configuration is not operational. **Example: set admin enabled** The default setting is **enabled**.
host-box *boxReference*	Specifies the master box on which the rule database runs and, optionally, backup boxes. See Master services description for more information. If there is no host box specified, SIP-based load balancing is not available to the system. Box 1 is the default box. Your first entry overwrites that default. Subsequent entries are added to the list as backup boxes. **Example: set host-box cluster box 2** The default setting is box **1**.

Property name	Description
group *groupID*	Associates the load balancing service with a VRRP group. Enter the number of a previously configured vrrp group. See Master services in VRRP configurations for a complete explanation. **Example: set group 1** Enter a value for the VRRP group. The default setting is **0** (no grouping association).
preempt {true | false}	*Secondary property.* Specifies whether the master service should retake the mastership if it has gone down and then returned to operation. If set to **true**, the master resumes its position. If set to **false**, the backup service retains master control. **Example: set preempt true** The default setting is **false**.
takeover-timer-value *milliseconds*	*Secondary property.* Specifies the number of milliseconds that the master service stays in “awaiting takeover” mode at boot time. When a box boots, each hosted master service waits for this period of time to determine if any existing boxes in the cluster are already running that service before assuming mastership. **Example: set takeover-timer-value 2000** The default setting is **1000** milliseconds.

file-mirror

Purpose

Sets all participating AA-SBC devices to share particular files (the types of files shared are preset in the AA-SBC operating system), such as media recordings, log files, etc. The file-mirror master-service distributes files to all devices listed as hosts for the service. It is used to make files highly available in the event that the box that created the file becomes unavailable. File mirroring includes keeping a record of each file in the file mirror database and also keeping a copy of each file on the local disk drive.

When configured, file mirroring works as follows:

1. When a file gets saved to the master file system, a record of the file is saved to the master database.
2. The master database then sends a message to all backup databases indicating a change and updating the backup.
3. The backup box(es) then compare their own database to their file system to determine if any files are missing (the new file is missing).
4. The backup then pulls the missing file(s) from the master file system.

Once the files are mirrored, you can play them back from any box that functions as a host. If accessing the file from a backup, the backup system first checks its database to make sure an entry is listed. It then checks its local disk for a copy of the file. If the file is not there (for example, an error during the pull operation) or is out of date, the backup again pulls the file from the master. File mirroring provides a secondary mechanism for assuring file availability. Non-host boxes also maintain a copy of the database and can pull files from the master as they are needed for processing. Use the file-mirror-service action to manage the mirrored files.

Syntax

```
config master-services file-mirror
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables file mirroring on the system. **Example: set admin enabled** The default setting is **enabled**.
host-box *boxReference*	Specifies the master box on which file mirroring is run and, optionally, backup boxes. See Master services description for more information. If there is no host box specified, there is no file mirroring. **Example: set host-box cluster box 1** There is no default setting.
group *groupID*	Associates the file mirroring process with a VRRP group. Enter the number of a previously configured vrrp group. See Master services in VRRP configurations for a complete explanation. **Example: set group 1** Enter a value for the VRRP group. The default setting is **0** (no grouping association).
file-mirror-directory *path*	Identifies the location of the directory on the local disk to which the system writes the files. Supply an absolute path name. **Example: set file-mirror-directory /cxc_common/mirror1** There is no default value.

Property name	Description
preempt {true | false}	*Secondary property.* Specifies whether the master service should retake the mastership if it has gone down and then returned to operation. If set to true, the master resumes its position. If set to false, the backup service retains master control. **Example: set preempt true** The default setting is **false**.
takeover-timer-value *milliseconds*	*Secondary property.* Specifies the number of milliseconds that the master service stays in “awaiting takeover” mode at boot time. When a box boots, each hosted master service waits for this period of time to determine if any existing boxes in the cluster are already running that service before assuming mastership. **Example: set takeover-timer-value 2000** The default setting is **1000** milliseconds.

external-backup

Purpose

Configures the system write all mirrored files to a backup server. With the specified frequency, the system writes all files contained in the file-mirror **file-mirror-directory** to the path specified. This is not a synchronization operation, it is strictly backup. Assuming the file transfer to the backup server was successful, the system does not attempt another transfer until the specified interval.

Syntax

```
config master-services file-mirror external-backup
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables the configuration for the off-box backup. **Example: set admin enabled** The default setting is **enabled**.
url *path*	Identifies the location of the backup server to which the system backs up the files. Supply an absolute path name. **Example: set file-mirror-directory /cxc_common/ mirror1** There is no default value.
refresh *minutes*	Specifies the frequency with which the system writes files to the backup server. See Setting time and time intervals for information on entry format requirements. **Example: set refresh 10:00** Enter a value between 1 minute and 14 days; the default setting is **15** minutes.

route-server

Purpose

Sets the route server master service, which manages the server process. The master service handles requests from local or remote AA-SBC devices or route server definitions. When presented with a request from the SIP process, the master service responds as follows, depending on the configuration:

- The master service retrieves one or more routes from the local (to the cluster) route server. This is the result if the session configuration authorization object is set to **Local**.
- The master service sends a Diameter request to retrieve the route(s) from the configured remote route server. This is the result if the session configuration authorization object is set to **Diameter**.

- The master service sends a request to an external policy service. This is the result if the session configuration authorization object is set to **WSDL**.

When multiple routes are returned, the dial-plan arbiter, if configured, resolves the best route.

The application can be configured in two ways—either intracluster or intercluster. Each has different configuration requirements, described below. Note that because AA-SBC propagates route server rate table updates to backup boxes, you do not need to configure the file-mirror service for it.

See *Net-Net OS-E – Session Services Configuration Guide* for information on installing and implementing the route server import client, a web application that imports routes into the database.

Intracluster route server

When two or more AA-SBC devices are within a cluster and the route server resides in the cluster, you can use intracluster route server. In that case, the route server lookup process is handled by AA-SBC within the cluster running this master service. To use intracluster routing, you must configure the following:

1. Set the primary and backup boxes that will host the **route-server** master service.
2. Enable the lookup destination by setting the **mode** property of the session configuration authorization object to **Local**.

Intercluster route server

When the route server for a cluster resides in a different cluster, use intercluster route server. Intercluster route server allows a remote system to serve as the route server for one or more clusters. To use intercluster routing, you must configure the following:

1. Configure a system in the local cluster as the **route-server** master service host (and optionally, configure backup boxes). The master is the system that handles local SIP requests for a route server lookup. The Diameter server configuration on this master determine where the request is forwarded.
2. Configure a system in the remote cluster to run the **route-server** master service, making it the server.
3. On the system in the remote cluster hosting the route server, configure the IP interface that listens for route server lookup (diameter) requests.

4. On the local system hosting the **route-server** master service, configure the list of diameter-groups to which this system should forward route server lookup requests.

5. Enable the lookup destination by setting the **mode** property of the session configuration authorization object to **Diameter**, and select a diameter-group from the list as the destination route server.

Syntax

```
config master-services route-server
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables the **route-server** master service on the system. **Example: set admin enabled** The default setting is **enabled**.
host-box *boxReference*	Specifies the master box on which route server is run and, optionally, backup boxes. See Master services description for more information. If there is no host box specified, there is no least-cost routing. **Example: set host-box cluster box 1** There is no default setting.
group *groupID*	Associates the route server process with a VRRP group. Enter the number of a previously configured vrrp group. See Master services in VRRP configurations for a complete explanation. **Example: set group 1** Enter a value for the VRRP group. The default setting is **0** (no grouping association).
preempt {true | false}	*Secondary property.* Specifies whether the master service should retake the mastership if it has gone down and then returned to operation. If set to true, the master resumes its position. If set to false, the backup service retains master control. **Example: set preempt true** The default setting is **false**.

Property name	Description
takeover-timer-value *milliseconds*	*Secondary property.* Specifies the number of milliseconds that the master service stays in “awaiting takeover” mode at boot time. When a box boots, each hosted master service waits for this period of time to determine if any existing boxes in the cluster are already running that service before assuming mastership. **Example: set takeover-timer-value 2000** The default setting is **1000** milliseconds.
max-routes {automatic | *integer*}	*Secondary property.* Specifies the maximum number of route entries that can be imported from the rate table to the route server database. The available range for this property is determined by license restrictions. **Example: set max-routes 1000000** The default setting is **automatic**.

sampling

Purpose

Opens the mechanism for setting the interval at which AA-SBC samples operational aspects of the system for, either, display in the AA-SBC Management System or for sending to an IBM Tivoli server. By setting sampling for a status provider, you can view data for that provider over a specified period of time. AA-SBC supports two sampling targets—a Postgres SQL database and an IBM tivoli server. Set the provider data sent to the target using the status and provider objects.

When you execute a status-provider command from the CLI, the system just displays the results of the request at the time it was issued. Once you have enabled sampling, the master service stores the samples in its local database. You can you can select a status provider underneath **Trends** in the **Status** tab when using the AA-SBC Management System. The GUI trends graphs pull data from the database on the sampling master service box to display a time series graph of the results. Changes to the interval setting in the sampling subobjects do not effect the CLI results.

➡ **Note:** If you have limited storage space, disable this feature if you are not using it. Otherwise, polling data is continuously written to the status database.

Configuring summary statistics for display

There are certain status providers that require the **sampling** master service in order to report data.For example, the configuration required to display data for the **show sip-summary-by-box** (and related, **show sip-summary-rates-by-box**) involves the following:

1. Enable this master service (sampling). This begins collection of the relevant data.
2. Add a database target.
3. Select the **sip-summary-by-box** provider. Set the **interval** for this provider to a very short period (minimum allowed is 30 seconds).

Syntax

```
config master-services sampling
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables status sampling on the system. In order to view the **Trends** graphs in the AA-SBC Management System, you must enable the sampling master service. **Example: set admin enabled** The default setting is **enabled**.
host-box *boxReference*	Specifies the master box on which status sampling is run and, optionally, backup boxes. See Master services description for more information. If there is no host box specified, there is no sampling. **Example: set host-box cluster box 1** There is no default setting.

Property name	**Description**
group *groupID*	Associates the status sampling process with a VRRP group. Enter the number of a previously configured vrrp group. See Master services in VRRP configurations for a complete explanation. **Example: set group 1** Enter a value for the VRRP group. The default setting is **0** (no grouping association).
preempt {true | false}	*Secondary property.* Specifies whether the master service should retake the mastership if it has gone down and then returned to operation. If set to true, the master resumes its position. If set to false, the backup service retains master control. **Example: set preempt true** The default setting is **false**.
takeover-timer-value *milliseconds*	*Secondary property.* Specifies the number of milliseconds that the master service stays in “awaiting takeover” mode at boot time. When a box boots, each hosted master service waits for this period of time to determine if any existing boxes in the cluster are already running that service before assuming mastership. **Example: set takeover-timer-value 2000** The default setting is **1000** milliseconds.

tivoli

Purpose

Configures AA-SBC to communicate with the IBM Tivoli server and to use it as a target for sampling data. (You can also send data to the local database.) The Tivoli server provide sampling status and event information to the IBM Tivoli Enterprise Manager. This information includes call information (e.g., call volumes and arrival rates, active calls, failed calls, etc.), network QoS (calculated Mean Opinion Score, Post-Dial Delay, Answer-Seize Ratio, Average Call Duration), and unit and cluster status (e.g., CPU usage, interface availability, etc.). The collected status and event information can be viewed and monitored using the IBM Tivoli Enterprise Portal desktop client.

Syntax

```
config master-services sampling tivoli hostName
```

Properties

Property name	Description
admin {enabled | disabled}	Sets the administrative status of the IBM Tivoli server configuration. **Example: set admin enabled** The default setting is **enabled**.
protocol {tcp | udp}	Specifies the protocol the Tivoli server uses to communicate with the system. **Example: set protocol udp** The default setting is **tcp**.
port *portNumber*	Specifies the port number over which the system communicates with the Tivoli server. **Example: set port 9650** The default setting is **7500**.

database

Purpose

Configures a database for status sampling collection. When enabled, the local database is a target for sampling data. Setting these properties has no effect on other database activities (i.e., log or system databases). This data is then available as a time-series graph in the **Trends** section of the AA-SBC Management System **Status** tab.

Syntax

```
config master-services sampling database
```

Properties

Property name	Description
admin {enabled | disabled}	Sets the administrative status of the local status database. When **enabled**, the system collects data from all status providers configured under this object. **Example: set admin enabled** The default setting is **enabled**.
duration {*days* | unlimited}	Specifies the number of days for which the system keeps collected provider data. **Example: set duration 5** The default setting is **7** days.

status

Purpose

Configures, through its subobjects, the status providers that report to the parent target. You can enable and disable the configuration through this object. There is one **status** object per target.

Syntax

```
config master-services sampling tivoli hostName status
config master-services sampling database status
```

Properties

None

provider

Purpose

Selects which providers to collect data from and defines the time period for which the system displays status provider statistics. This data is then available as a time-series graph in the Trends section of the AA-SBC Management System **Status** tab (if the target is database) or sent to the tivoli server. The following table describes each provider available for configuration through the status object and, if applicable, the Trend graph in which the system displays the information. The property options are the same for each.

provider object (show command)	**Reports...**	**Trend graph (database target only)**
interface-details	...packet and error rates for each configured Ethernet interface.	interface-details
interface-throughput	...interface throughput statistics over various time intervals.	interface-throughput
system-heap	...usage and failure rates for each process running on the box.	system-heap-summary
trunk-groups	...carrier, exchange, server, trunk, as well as usage statistics.	tivoli only
server-load-db	...the peer, associated server, and aggregate load for each server that participates in load balancing.	tivoli only

provider object (show command)	Reports...	Trend graph (database target only)
switch-pool	...gateway(s) and associated swith(es), as well as configuration and usage statistics.	tivoli only
cpu-usage	...CPU utilization statistics over various time intervals.	cpu-usage
im-content-counters	...the number of times each word list or URL list has triggered a policy.	im-content-counters
sip-summary-by-box	....counters and values relevant to the SIP process. This is a base status class that is used by other status providers to calculate differential counters. See Configuring summary statistics for display for configuration information.	N/A
sip-stack	...packets sent and received. Also displays total calls (both admitted and rejected) and active calls.	call-count packet-count
call-admission-control	...current and peak calls, calls in flight, active calls dropped, and in-flight calls dropped.	current-call-admission-control max-call-admission-control
tls-admission-control	...current TLS calls, as well as peak statistics for TLS calls, calls dropped, calls in-flight and, in-flight calls dropped.	tls-call-admission-control
active-calls	...maximum and average session duration.	session-duration
registration-status	...total received and declined registrations.	registrations

provider object (show command)	Reports...	Trend graph (database target only)
location-auth-summary	...the number of registrations aborted (AORs challenged, those delegated to an upstream server due to lack of SIP server response, and those AORs initially unable to register because of too many concurrent requests).	registrations-aborted
location-reject-summary	...the number of registrations that were discarded, declined, or rejected.	registrations-rejected
dos-transport-counters	...the number of times the DOS transport policy has been triggered.	dos-transport-counters
dos-sip-counters	...the number of times the DOS SIP policy has been triggered.	dos-sip-counters
dos-url-counters	...the number of times the DOS URL policy has been triggered.	dos-url-counters

Syntax

```
config master-services sampling tivoli hostName status provider
config master-services sampling database status provider
```

Properties

Property name	Description
admin {enabled | disabled}	Specifies whether CPU usage statistics are stored for the specified provider in the target. **Example: set admin enabled** The default setting is **disabled**.
interval [*days* days] *hh:mm:ss*	Defines how often the system polls the status provider for data. Once queried, the system writes the data to the status database. The data is kept (and displayed) for the amount of time set in the **duration** property of the target configuration. The interval property is required. The system interprets the interval you enter from right to left, allowing you to enter only part of the time string. For example: • 30=30 seconds • 1:30=90 seconds • 1:00:00=one hour To enter a number of days, enter the number and the keyword days, and optionally, the time string. Enclose the entry in quotation marks. **Example: set interval "5 days 12:00:00"** The default interval is **1:00:00** (one hour).

jtapi

Purpose

Enables integration between Microsoft OCS and third-party call control (3PCC) servers (e.g., BroadWorks, Cisco Call Manager and Avaya AES). AA-SBC communicates with OCS using CSTA-over-SIP. (CSTA is a protocol used by OCS to communicate call state information, which can then be reflected in user presence status.) AA-SBC communicates with the 3PCC servers using OCI (BroadWorks) or JTAPI (Cisco and Avaya). See Chapter 71, "Third-party call control server objects", for information on configuring the servers.

Syntax

```
config master-services jtapi
```

Properties

Property name	Description
admin {enabled | disabled}	Enables and disables the system third-party call control master service. **Example: set admin enabled** The default setting is **enabled**.
host-box *boxReference*	Specifies the master box on which 3PCC services run and, optionally, backup boxes. See Master services description for more information. If there is no host box specified, no directory services can run. **Example: set host-box cluster box 2** There is no default setting.
group *groupID*	Associates the 3PCC service with a VRRP group. Enter the number of a previously configured vrrp group. See Master services in VRRP configurations for a complete explanation. **Example: set group 1** Enter a value for the VRRP group. The default setting is **0** (no grouping association).

Property name	Description
preempt {true | false}	*Secondary property.* Specifies whether the master service should retake the mastership if it has gone down and then returned to operation. If set to true, the master resumes its position. If set to false, the backup service retains master control. **Example: set preempt true** The default setting is **false**.
takeover-timer-value *milliseconds*	*Secondary property.* Specifies the number of milliseconds that the master service stays in “awaiting takeover” mode at boot time. When a box boots, each hosted master service waits for this period of time to determine if any existing boxes in the cluster are already running that service before assuming mastership. **Example: set takeover-timer-value 2000** The default setting is **1000** milliseconds.

available-memory

Purpose

The available-memory object allows you to enable a sampling interval for the AA-SBC to check the available memory.

Syntax

```
config master-services sampling database status available-memory
```

Properties

Property name	Description
admin [enabled | disabled]	Enable or disable the AA-SBC checking the available memory. **Example: set admin disabled** The default setting is **enabled**.
interval	Defines how often the AA-SBC polls the status provider for data. **Example: set interval 10** Min: 30 / Max: 1036800 The default setting is **1:00:00** (1 hour).

40. Media ports object

Media port pool description

The media port pool defines the IP addresses and port ranges to assign to media streams on an Ethernet interface. These ports are used by the AA-SBC media services (media anchoring, NAT, recording, etc.), which are configured in the (default-) session-config media object.

Media ports object summary

The following table lists and briefly describes the **media-ports** objects. See the following chapters for other objects in the CLI hierarchy:

- Chapter 14, “Cluster, box, and interface objects”
- Chapter 77, “VLAN objects”
- Chapter 35, “IP objects”

Object name	Description
`media-ports`	Configures the media port pool on this system interface.

media-ports

Purpose

Configures the media port pool on this interface, defining the available addresses and ports to use for media services. These are the services defined in the session configuration media object. These ports can be used for media distribution, if the cluster object **share-media-port** property is set to true. Use the **show media-ports-process-units** command to view port limits and configuration for each processor. Use the port-limit object to define thresholds for active media port use.

You can configure AA-SBC to use ports on the **cxc** processor. Do not configure media ports on AA-SBC ports (typically eth0 through eth3).

Syntax

```
config cluster box number interface ethX ip name media-ports
config cluster box number interface ethX vlan number ip name
   media-ports
config box interface ethX ip name media-ports
config box interface ethX vlan number ip name media-ports
```

Properties

Property name	Description
admin {enable | disabled}	Enables or disables the media port pool on this system interface. **Example: set admin disabled** The default setting is **enabled**.
base-port *integer*	Sets the base or starting port number to use for this port pool. **Example: set base-port 20000** Enter a value from 1 to 65535. The default setting is **20000**.

Property name	**Description**
count *integer*	Sets the total number of ports available for the media port pool. **Example: set count 2000** Enter a value from 0 to 65535. The default setting is **5000**. If you set the value to 0, the pool is empty, which is the equivalent of disabling the pool.
idle-monitor {enabled | disabled}	Enables or disabled the monitoring of idle ports by AA-SBC. When ports are not in use, they should not receive traffic. When **enabled**, this property ensures that no traffic is sent to idle ports that are part of the media pool. If the system detects that an idle port is receiving traffic, the port is put into a quarantine list. An internal timer releases the port when traffic to the port stops for a period of seconds. **Example: set idle-monitor disabled** The default setting is **enabled**.

41. Messaging objects

Messaging description

Messaging is the mechanism AA-SBC uses to communicate among boxes. By configuring messaging, you are setting up a listening socket on an interface. This enables the interface to receive messaging traffic and participate in clustering and media partnering.

In clustering, the master box in a cluster looks through the configurations of all drones to find which interface each drone is using for messaging. (If multiple interfaces are configured, the master only communicates with one—the first it finds.) The master then communicates with the identified interface to share configuration and data. See Chapter 14, “Cluster, box, and interface objects” for more information.

In media partnering, you configure a specific IP address (on a different box) as a partner. On the box that houses that address, you would need to configure and enable messaging in order for partnering to work. See the cluster media-partners object for more information.

Messaging object summary

The following table lists and briefly describes the **messaging** object. See the following chapters for other objects in the CLI hierarchy:

- Chapter 14, “Cluster, box, and interface objects”
- Chapter 77, “VLAN objects”
- Chapter 35, “IP objects”

Object name	Description
`messaging`	Configures and enables messaging for an interface.

messaging

Purpose

Configures and enables messaging for the specified interface. Messaging provides the mechanism for the AA-SBC media partners and clustering capabilities.

Note: The protocol, port, and certificate set within this object must match the values set for these properties in the cluster media-partner partner object. If they do not match, the systems will not be able to communicate.

Syntax

```
config cluster box number interface ethX ip name messaging
config cluster box number interface ethX vlan number ip name messaging
config box interface ethX ip name messaging
config box interface ethX vlan number ip name messaging
```

Properties

Property name	Description
admin {enabled | disabled}	Sets the administrative state of the messaging configuration. When **disabled**, the parameters of messaging can still be configured, but the interface cannot participate in media partnering or clustering. **Example: set admin disabled** The default setting is **enabled**.
protocol {TCP | TLS}	Specifies the protocol the interface uses to communicate between systems. **Example: set protocol TLS** The default protocol is **TCP**.

Property name	Description
port *number*	Identifies the Ethernet port through which the system listens for messaging sessions. **Example: set port 13333** Enter a port number between 1 and 65535. The default Telnet port is **5132**.
certificate *certificateReference*	Assigns the certificate that must be presented to participate in message exchanges if TLS is used as the protocol. Enter a reference to a previously configured certificate. **Example:** **`set certificate vsp tls certificate nnos-e.companyA.com`** **`There is no default setting.`**

42. Monitor-group objects

Monitor-group description

The monitor-group configuration allows you to configure an endpoint, other than the call recipient, that can listen in on a call. For example, you can forward an active SIP call directed to a 911 emergency number to another SIP phone so that a third party, such as a supervisor, can listen in on the active call. This is called snooping. Note that this feature differs from the session configuration handle-response three-way calling feature. When using call monitoring, a third-party can join a conversation.

Once the monitor-group object is configured, you can associate it with a policy using the session-config media object **monitor** property.

In addition, if you also select to record the call you can use the AA-SBC Management System **Call Logs** tab to play the recorded SIP sessions. For complete information on configuring playback groups and policy association, refer to the *Net-Net OS-E – Session Services Configuration Guide*.

Monitor-group object summary

The following table lists and briefly describes the **monitor-group** objects. See the following chapter for other objects in the CLI hierarchy:

- Chapter 75, “Virtual system partition (VSP) objects”

Object name	Description
`monitor-group`	Opens or creates the specified monitor-group configuration object on the system.
`monitor-endpoint`	Sets the target URI to which the system also forwards the call.

monitor-group

Purpose

Opens the monitor group object for editing. The monitor-group configuration allows you to forward calls to a configured endpoint. The endpoint configured specifies the URI location from which a third party can listen-in on the active call.

➡ **Note:** You must enable media anchoring for listening in on a SIP session. Do this using the session configuration media object **anchoring** property. Anchoring forces a SIP call request to be handled by the AA-SBC proxy, (residing between the caller and the call destination) where the call sharing, recording, and playback mechanisms are available.

Syntax

```
config vsp monitor-group name
```

Properties

Property name	Description
`admin {enabled | disabled}`	Enables or disables the system monitor-group configuration. When **enabled**, the system forwards the SIP call to a target phone (endpoint). Configure the target using the monitor-endpoint object. **Example:** set admin enabled The default setting is **enabled**.

monitor-endpoint

Purpose

Sets the destination endpoint where SIP calls can be listened in on for review or security purposes.

The call playback-endpoint specifies the target SIP phone from which you can monitor a SIP call session while the call is in progress. Once the SIP session is established, AA-SBC rings the endpoint SIP phone from where a listener can monitor the call. (There is slight time delay between actual session and the playback endpoint.). The user at the call playback endpoint can even terminate the RTP session between the SIP recipients, if necessary.

Syntax

```
config vsp monitor-group name monitor-endpoint name
```

Properties

Property name	**Description**
to *URI*	Sets the target SIP address of the endpoint that can monitor the call. This is the address to which the system forwards the call. **Example: set to sip:supervisor@Company.com** There is no default setting.
from *URI*	Configures the From field of the SIP header. This is an informational field that would display, for example, to identify to the person monitoring the call that the call was from the system. **Example: set from sip:nnos-e-monitor@companyA.com** There is no default setting.
transport {any | UDP | TCP | TLS}	Sets the default protocol over which the SIP call session is forwarded to the destination SIP server. **Example: set transport TLS** The default setting is **any**.

43. Near-side NAT object

Near-side NAT description

You can configure AA-SBC to perform address translation on behalf of an enterprise firewall device. To do so, configure the parameters of the **near-side-nat** object to match the settings of your firewall and your media and SIP ports.

AA-SBC uses network address translation (NAT) to change its private, backend address(es) to a public, routable address. NAT is defined in *RFC 1631, The IP Network Address Translator*. NAT ensures that internal private network addresses are rewritten so that they appear to come from the designated external network address. AA-SBC modifies outgoing packets so that the return address is a valid Internet host (the firewall). The firewall then changes the destination address on incoming packets to the AA-SBC private address. This process protects the private addresses from public view. In addition, because the private address is not routable, any returning packets would not reach their destination. NAT provides a routable address through which AA-SBC can maintain SIP and media connections.

AA-SBC works with the firewall as follows:

1. You configure your firewall appropriately.
2. Configure AA-SBC to match the firewall settings for IP addresses and ports.
3. Configure the AA-SBC **ip** sip and **ip** media-ports ports to recognize the ports specified in this object.
4. When AA-SBC detects a packet coming from the firewall over the UDP or TCP listening port, it performs address translation where AA-SBC changes the source address on outgoing packets from its own internal IP to the firewall public-facing address (set with the **public-ip** property).

See the following chapters for related information:

- Chapter 65, “Session Initiation Protocol objects”
- Chapter 40, “Media ports object”

Near-side NAT object summary

The following table lists and briefly describes the **near-side-nat** object. See the following chapters for other objects in the CLI hierarchy:

- Chapter 14, “Cluster, box, and interface objects”
- Chapter 77, “VLAN objects”
- Chapter 35, “IP objects”.

Object name	Description
near-side-nat	Configures the NAT settings between AA-SBC and the external firewall.

near-side-nat

Purpose

Creates or edits a firewall configuration that allows AA-SBC to perform address replacement for packets originating from AA-SBC and destined for the Internet. By configuring the IP address of the public-facing interface on the enterprise firewall, AA-SBC can produce a contact header that replaces its own, private IP address with the public-facing address of the firewall to allow completion of SIP calls.

When configuring AA-SBC for address replacement, you must mirror the port forwarding of the firewall. The ports that you configure within this object indicate to AA-SBC when it should do address replacement. For example, if you have configured the UDP port at 5060, when AA-SBC receives a packet from the firewall device using port 5060, it will replace its private IP address with the configured public address in its response.

Enter a name for the firewall configuration when opening this object.

Configuration requirements

You typically configure UDP port 5060 and TCP ports 5060 and 5061 for SIP traffic. Be certain that the port numbers you enter here are the same as those you configured in the **ip** sip object.

In addition, you may configure the NAT pool addresses within this object, typically UDP ports 20000 through 30000. Be certain that the port numbers you enter here are the same as those you configured in the **ip** media-ports object.

Finally, the port numbers and IP address that you specify must match the configuration on your external firewall device.

Syntax

```
config cluster box number interface ethX ip name near-side-nat name
config cluster box number interface ethX vlan number ip name
   near-side-nat name
config box interface ethX ip name near-side-nat name
config box interface ethX vlan number ip name near-side-nat name
```

Properties

Property name	Description
admin {enabled | disabled}	Sets the administrative state of the external firewall configuration on the system, either **enabled** (active) or **disabled**. When disabled, you can still configure the firewall parameters, but the system will not do the address replacement necessary (if it is situated behind a near-side firewall). **Example: `set admin enabled`** The default setting is **enabled**.
public-ip *ipAddress*	Sets the public-side address of the firewall positioned between the system and the Internet. The system will replace its own private, internal network address with the firewall public-facing IP address. **Example: `set public-ip 1.2.3.4`** There is no default IP address. You must supply a globally unique, routable value.

Property name	Description
udp-range *startingPort count*	Specifies the UDP port number(s) that the system is listening for packets from the firewall device on. When the system receives a packet from the specified firewall port, it executes an address replacement of its own private address with the firewall public address in its response. The typical SIP port is 5060; the typical media pool range is 20000 through 30000. Make sure that your UDP port configuration matches both the firewall configuration and the **ip** sip and **ip** media-ports object configurations. If no UDP port is specified, the system has no port to listen for, and therefore, no address replacement occurs. **Example: `set udp-range 5060 1`** There is no default UDP range.
tcp-range *startingPort count*	Specifies the TCP port numbers that the system is listening for packets from the firewall device on. When the system receives a packet from the specified firewall port, it executes an address replacement of its own private address with the firewall public address in its response. The typical SIP port is 5060 and 5061. Make sure that your TCP port configuration matches both the firewall configuration and the **ip** sip object configuration. If no TCP port is specified, the system has no port to listen for, and therefore, no address replacement occurs. **Example: `set tcp-range 5060 2`** There is no default TCP range.

44. NTP client and server objects

NTP description

AA-SBC system uses Network Time Protocol (NTP), Version 4 (described in RFC 1305) to synchronize its clock with network clocks. Systems using NTP all set and maintain their internal clock to the Coordinated Universal Time (UTC).

Synchronized time across the network provides packet and event time stamps and security certificate validation. Because computer clocks drift a few seconds a day, networked systems can be out of synchronization. NTP uses time signals from accurate time sources on the Internet to ensure all systems are synchronized.

Each AA-SBC device has a real time clock that uses NTP to maintain accurate time. You can configure AA-SBC as an NTP client and/or as an NTP server. An NTP server responds to NTP client requests, using either external Internet time sources or the time set manually via the CLI, depending on the configuration. An NTP client queries the configured time server at the specified interval.

Note: The NTP client and server objects are located in different places in the CLI hierarchy. You configure the client within the box object and the server within the interface object.

NTP client object summary

You configure NTP clients from the box and cluster configuration objects. The box configuration object allows you to configure the NTP client on the locally attached AA-SBC; the cluster configuration object allows you to configure the NTP client on individually numbered (indexed) AA-SBC devices in a network cluster.

The following table lists and briefly describes the **ntp-client** objects. See the following chapters for other objects in the CLI hierarchy:

- Chapter 14, “Cluster, box, and interface objects”

Object name	Description
`ntp-client`	Opens the ntp-client configuration object. for editing.

NTP server object summary

You configure NTP servers on AA-SBC Ethernet and VLAN interfaces.

The following table lists and briefly describes the `ntp-server` objects. See the following chapters for other objects in the CLI hierarchy:

- Chapter 14, “Cluster, box, and interface objects”
- Chapter 77, “VLAN objects”
- Chapter 35, “IP objects”

Object name	Description
`ntp-server`	Opens the ntp-server configuration object for editing.

ntp-client

Purpose

Opens the NTP client configuration object on the locally attached AA-SBC device, or opens the NTP client configuration object on the specified AA-SBC device in a cluster configuration.

Syntax

```
config box ntp-client
config cluster box integer ntp-client
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables NTP client services on the local system. **Example: set admin enabled** The default setting is **enabled**.
server *ipAddress*	Specifies the IP address of the NTP server. This is the address of the device that responds to this NTP client requests. Enter the IP address in dotted decimal format or a host name. **Example: set server 192.168.10.10** There is no default setting.
poll-interval *minutes*	Sets the poll interval for NTP updates. The interval is the number of minutes between NTP client requests to the NTP server. **Example: set poll-interval 5** Enter a value between 1 and 1440. The default poll-interval is **10** minutes.

ntp-server

Purpose

Opens the NTP server configuration object on the specified AA-SBC Ethernet or VLAN interface. When you enable NTP server functionality, the interface will respond to NTP client requests with its current time. To set the time on the box, do one of the following:

- use the clock action.
- set the server as a client to an external Internet time server using the ntp-client object.

Syntax

```
config box interface ethX ip name ntp-server
config cluster box integer interface ethX ip name ntp-server
```

Properties

Property name	Description
`admin {enabled | disabled}`	Enables or disables the NTP server on the current system Ethernet or VLAN interface. The default setting is **enabled**.

45. OCI settings objects

OCI settings description

BroadWorks servers provide a Client Application Protocol Open Client Interface (CAPOCI). This interface is similar to CSTA for SIP Phone User Agents (uaCSTA), which allows CSTA to provide a subset of CSTA call control functionality, called first-party call control, for SIP user agents. AA-SBC supports a translation between these two, allowing use of CSTA for communication with LCS servers and CAP OCI for communication with BroadWorks servers.

The CSTA properties themselves are set within policy, using the csta-settings object. For more information on CSTA, see *Standard ECMA-269, Services for Computer Supported Telecommunications Applications (CSTA) Phase III.*

The **oci-settings** object defines the properties of the AA-SBC interface that interacts with the BroadWorks server.

OCI settings object summary

The following table lists and briefly describes the **oci-settings** object. See the following chapter for other objects in the CLI hierarchy:

- Chapter 75, “Virtual system partition (VSP) objects”

Object name	Description
`oci-settings`	Configures the properties of the interaction with the OCI server.

oci-settings

Purpose

Configures the properties of the AA-SBC interface that interacts with the BroadWorks server. The interface itself is configured with the csta-settings object.

Syntax

```
config vsp oci-settings
```

Properties

Property name	Description
remote-number-normalization {enabled [*extensionLength*] | disabled}	Specifies whether to add the digits to the incoming phone number. Use this if the system does not have the full number, for example, during intercompany dialing that only uses extensions. When **enabled**, the system completes the number with the monitored phone number less the number of digits specified. For example, with the extension digits set to four, the system adds the monitored phone number (excluding the last digits) to the incoming number. When **disabled**, the number is left unchanged. **Example: set remote-number-normalization disabled** The default setting is **enabled**, with an extension length of **4**.
conference-delay *milliseconds*	Specifies the number of milliseconds delay before converting a CSTA Conference Call request to the format required for an OCI conference request. **Example: set conference-delay 1000** The default setting is **500** milliseconds.

Property name	Description
local-phone-registration {true | false}	Specifies whether the originating (monitored) phones are registered through the system. If set to **true**, OC clients that are using CSTA must register with the system to receive the additional CSTA features. If set to **false**, the phones do not have access to the features, regardless of whether they are registered with the system. **Example: set local-phone-registration false** The default setting is **true**.
worker-threads *threads*	Specifies the number of threads the system dedicates to processing and responding to OCI messages from the server. **Example: set worker-threads 5** Enter a value between 1 and 10; the default setting is **2** threads per connection.
tcp-keepalives {enabled | disabled}	Specifies whether the system sends TCP keepalive messages to the OCI server. If enabled, the system sends regular keepalives and times out the connection if they go unanswered. The frequency and timeout interval are defined in the TCP RFC. **Example: set tcp-keepalives enabled** The default setting is **disabled**.
oci-keepalive-timeout *minutes*	Specifies the frequency with which the system sends OCI keepalive messages to the server. If the system does not receive any traffic from the server (the connection has been idle), it sends a keepalive so that the server does not declare the connection down. **Example: set oci-keepalive-timeout 7** Enter a value from 1 to 30; the default setting is **5** minutes.

OCI settings objects

46. Outbound and inbound normalization objects

Normalization is the process of making modifications to the SIP URI in order to conform to the requirements of an upstream server and to simplify lookups. (Normalization occurs before arbitration or routing lookup.) By normalizing the URI, you simplify, for example, lookups, policy matches, and user service matches.

Normalization in the configuration

The normalization objects are available in dial plans, registration plans, servers, and carrier exchanges and trunk groups. Dial and registration plan normalization is handled in one way, based on a specific model. Their normalization options remain in those chapters. Servers and carriers use a different model. They share a method of applying normalization (described in this chapter), but that method differs from the dial or registration plan method. Locations for normalization configuration are defined in the following table.

Path	Defines...
vsp dial-plan normalization	Normalization of INVITE messages; applicable for endpoints. See Chapter 21, “Dial plan objects”, for more information.
vsp registration-plan normalization	Normalization of REGISTER messages; applicable for endpoints. See Chapter 55, “Registration plan objects”, for more information.
vsp enterprise servers *serverType name* server-pool server *name*	Normalization of selectable header and message types coming in and out of a server (described in this chapter). See Chapter 59, “Server objects”, for all other server object configuration information.

Path	Defines...
vsp carriers carrier *name* exchange *name* switch *name*	Normalization of selectable header and message types coming in and out of a gateway (described in this chapter). See Chapter 12, “Carriers objects”, for all other gateway object configuration information.
vsp carriers carrier *name* exchange *name* switch *name* trunk-group *name*	Normalization of selectable header and message types coming in and out of a trunk group (described in this chapter). See Chapter 12, “Carriers objects”, for all other trunk group object configuration information.

Normalization description

Normalization works by identifying which messages are subject to normalization:

1. The **match** property sets the match for criteria SIP messages.
2. The **apply-to-header** property sets the header type(s) to which AA-SBC applies the **match** property.
3. The **apply-to-method** property sets the message type(s) to which AA-SBC applies the **match** property.

AA-SBC then applies changes to the USER field of the Request, To, and From URI. For outbound normalization, you can optionally apply more fine-grained modifications to the URI through policy application.

AA-SBC uses a longest-prefix match lookup to match the most specific entry. However, a SIP URI can match more than one pattern. See Chapter 21, “Dial plan objects” for information on:

- Pattern match precedence (important for correctly configuring normalization)
- Understanding phone synchronization

Normalization model for servers, gateways, and trunk groups

AA-SBC allows configuration of both inbound and outbound normalization plans. Inbound normalization applies to messages received from the server. Outbound applies to messages sent to the server.

Outbound and inbound normalization objects

Common properties

Both inbound and outbound normalization share some common properties. These are the matching criteria and the applied USER URI changes. Each is described below. In addition, the outbound normalization scheme allows altering URIs to make them acceptable to the upstream server.

User URI normalization

The **request-user**, **to-user**, and **from-user** normalization properties all support the same settings. See the property descriptions for an explanation of which is the effected portion of the URI for that property. These settings are described in the following table:

Setting	Description
`no`	No normalization applies, the request URI remains unchanged.
`prepend` *`phonePrefix`*	Adds the specified phone prefix to the beginning of the current phone number.
`prepend-to` *`resultingStringLength phonePrefix`*	Adds the specified phone prefix to the beginning (portion left of the @ sign) of the URI. Specify the prefix and the resulting string length, which indicates how many total characters are in the phone number after AA-SBC prepends the phone prefix.
`strip-off` *`numberOfCharacters`*	Removes as many characters as you specify from the phone prefix. Characters are removed beginning at the far left, moving towards the @ sign.
`strip-off-to` *`resultingStringLength`*	Shortens the phone prefix (portion left of the @ sign) to the number of characters you specify as the resulting string length.
`replace-prefix` *`newPhonePrefix`*	Replaces the prefix (left-most characters) with the characters for the new phone prefix that you supply. For example, if you supply three characters, AA-SBC will replace the first three characters with those you specify.
`replace-with` *`newPhoneNumber`*	Replaces all numbers to the left of the @ sign with the number you specify.

Setting	Description
`append` *`phoneExtension`*	Appends the specified extension to the end of the current phone number.
`thru-registration-plan`	Uses the normalization settings from the VSP's registration plan (defined in the registration-plan object).

The following table provides examples of how AA-SBC prepends prefixes in various situations.

Start with...	Changes to...	Explanation
Property setting: set from-user prepend-to 10 978		
4321	9780004321	AA-SBC expands the space between the prefix and the original number with 0s.
7654321	9787654321	AA-SBC prepends the prefix, the resulting string length is correct, so no further changes are made.
555557654321	9787654321	AA-SBC prepends the prefix (3 digits), and then includes the necessary number of digits to make the resulting string length (7), starting at the @ sign and moving left. AA-SBC removes all other digits.
Property setting: set from-user prepend-to 10 3219876543210		
4321	9876543210	AA-SBC prepends as many digits as specified by the resulting string length, beginning at the @ sign and moving left. AA-SBC removes all other digits.

Altering URIs

In some cases, it is necessary to change a part of the URI (outbound only) so that the next-hop server can accept the SIP message when it arrives. For example, a server may require a part of the URI to be in a specific format. AA-SBC allows you to modify the Host portion of the REQUEST, TO, and/or FROM URI so that the header matches any necessary requirements.

When you select to alter the URI, you can set AA-SBC to replace the specified (typically the HOST) field with one of the following:

Outbound and inbound normalization objects

- **none**—the URI is not modified. This is the default setting.
- **next-hop-ip**—the next-hop IP address, which is specified in the **host** property of:
  - server—server-pool server object
  - gateway—server-pool server object
  - trunk-group—switch object
- **next-hop-domain**—the next-hop domain, which is:
  - server—**peer-domain** property of the sip-gateway server object
  - gateway—**domain** property of the carrier object
  - trunk-group—**domain** property of the carrier object
- **local-ip**—the IP address of the interface the packet goes out on.
- **host** *string*—the IP address or host name that you specify.
- **directory** *directoryReference*—a user alias. When selected, AA-SBC looks for all aliases associated with the user listed in the To, Request, and From fields of the URI. AA-SBC then uses the alias associated with the referenced directory.

For example, it is not uncommon for a carrier to change a user ANI (automatic number identification) to the modified number used by the DNIS (dialed number identification service). For AA-SBC to correctly forward the call, it must put the user ANI back into the From header. It can do this if you configure the header-settings object to a referenced directory that identifies the DNIS alias.

Normalization object summary

The following table lists and briefly describes the normalization configuration objects. See the following chapter for other objects in the CLI hierarchy:

- Chapter 75, “Virtual system partition (VSP) objects”
- Chapter 21, “Dial plan objects”
- Chapter 59, “Server objects”
- Chapter 12, “Carriers objects”

Object name	Description
outbound-normalization	Applies normalization settings to outbound calls through this trunk group.
inbound-normalization	Applies normalization settings to inbound calls through this trunk group.

outbound-normalization

Purpose

Defines how AA-SBC manipulates headers for SIP messages directed to the server, gateway, or trunk-group. These settings effect how the call is routed to that destination, and are applied to the forking endpoint. Outbound normalization facilitates processing of SIP messages from AA-SBC for the next hop server/gateway by changing the messages so that they can conform to the requirements of the upstream server.

The outbound normalization plan has properties (alter-uri) that can change the Host portion of the selected URI field. To change more than the Host portion, use either:

- **outbound-session-config-entry** property of the sip-gateway server object
- **session-config-pool-entry** property of the server-pool server outbound-normalization object.

Syntax

```
config vsp enterprise servers server name server-pool server name
    outbound-normalization name
config vsp carriers carrier name gateway name outbound-normalization
    name
config vsp carriers carrier name gateway name trunk-group name
    outbound-normalization name
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables this normalization plan. When **enabled**, the system provides normalization for matching SIP messages. When **disabled**, you can configure the plan properties but the system does not apply it. **Example: set admin enabled** The default setting is **enabled**.
uri-match *type*	Specifies the match criteria SIP messages. Calls matching the criteria defined are then normalized. The **apply-to-headers** property defines which headers have the criteria applied for matching. **Example: set uri-match phone-prefix 978823** There is no default setting.
condition-list-match-secondary {false | true}	Specifies whether a condition list match should also be required for the specified **uri-match** property. If the match property is set to something other than condition-list, you can set this property to **true** to use a condition list in addition to the type selected. In that case, the call must match both the primary key and the condition list. **Example: set condition-list-match-secondary true** The default setting is **false**.
apply-to-headers {request-uri | to-header | from-header}	Specifies the header type to which the system applies the **uri-match** property. Headers containing the configured match in the selected method(s) are then normalized according to this plan. You can enter multiple header types; separate them by a plus sign (+) with no spaces. **Example: set apply-to-headers request-uri+from-header+ to-header** The default setting is **request-uri**.

Property name	Description
priority *value*	Specifies an order of preference for this outbound normalization plan. Often, a number or URI will match multiple normalization entries. By default, the system uses the most specific match. Use this property to override that default behavior and set a preference based on the **uri-match**. **Example: set priority 50** Enter any number greater than 1; the default priority setting is **100**.
apply-to-methods *messageType*	Specifies the message type to which the system applies the **uri-match** property. Those messages containing the configured match in the selected header(s) are then normalized according to this plan. When you modify this value, the system overwrites the current setting with only the message types you specify. For example, if set to the default (multiple selected) and you enter **OPTIONS**, the system will apply normalization only to the Options portion of the header. Enter multiple types separated by a plus sign (+) with no spaces. **Example: set apply-to-methods INVITE+REFER** The default setting applies normalization to the **INVITE+REFER+MESSAGE+INFO+OPTIONS+REGISTER +SUBSCRIBE+NOTIFY+PUBLISH** methods.
request-user *setting*	Sets the type of phone number manipulation that the system applies to outgoing calls to a server (to the USER field of the Request URI). See User URI normalization for property setting options and descriptions. **Example: set request-user prepend 1978** The default type setting is **no**.
to-user *setting*	Sets the type of phone number manipulation that the system applies to outgoing calls to a server (to the USER field of the To URI). See User URI normalization for property setting options and descriptions. **Example: set to-user strip-off-to 10** The default type setting is **no**.

Outbound and inbound normalization objects

Property name	Description
from-user *setting*	Sets the type of phone number manipulation that the system applies to outgoing calls to a server (to the USER field of the From URI). See User URI normalization for property setting options and descriptions. **Example: set from-user thru-registration-plan** The default type setting is **no**.
session-config-pool-entry *sessionConfigReference*	Specifies a previously configured entry in the session-config-pool object. If this property is set, the system applies the session configuration characteristics to all calls matching this outbound normalization plan. **Example: set vsp session-config-pool-entry entry 1** There is no session configuration applied by default.

inbound-normalization

Purpose

Defines how AA-SBC manipulates headers for SIP messages received from the server, gateway, or trunk-group. Inbound normalization facilitates AA-SBC dial-plan and location cache lookup to enforce policies and to forward calls and registrations.

Syntax

```
config vsp enterprise servers server name server-pool server name
   outbound-normalization name
config vsp carriers carrier name gateway name outbound-normalization
   name
config vsp carriers carrier name gateway name trunk-group name
   outbound-normalization name
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables this normalization plan. When **enabled**, the system provides normalization for matching SIP messages. When **disabled**, you can configure the plan properties but the system does not apply it. **Example: set admin enabled** The default setting is **enabled**.
uri-match *type*	Specifies the match criteria for SIP messages. Calls received from this server and matching the criteria defined are then normalized. The **apply-to-headers** property defines which headers have the criteria applied for matching. **Example: set uri-match phone-prefix 978823** There is no default setting.
condition-list-match-secondary {false | true}	Specifies whether a condition list match should also be required for the specified **uri-match** property. If the match property is set to something other than condition-list, you can set this property to **true** to use a condition list in addition to the type selected. In that case, the call must match both the primary key and the condition list. **Example: set condition-list-match-secondary true** The default setting is **false**.
apply-to-headers {request-uri | to-header | from-header}	Specifies the header type to which the system applies the **uri-match** property. Headers containing the configured match in the selected method(s) are then normalized according to this plan. You can enter multiple header types; separate them by a plus sign (+) with no spaces. **Example: set apply-to-headers request-uri+from-header+to-header** The default setting is **request-uri**.

Property name	Description
priority *value*	Specifies an order of preference for this normalization plan entry. Often, a number or URI will match multiple normalization entries. By default, the system uses the most specific match. Use this property to override that default behavior and set a preference based on the **uri-match**. **Example: set priority 50** Enter any number greater than 1; the default priority setting is **100**.
apply-to-methods *messageType*	Specifies the message type to which the system applies the **uri-match** property. Those messages containing the configured match in the selected header(s) are then normalized according to this plan. When you modify this value, the system overwrites the current setting with only the message types you specify. For example, if set to the default (all selected) and you enter **OPTIONS**, the system will apply normalization only to the Options portion of the header. Enter multiple types separated by a plus sign (+) with no spaces. **Example: set apply-to-methods INVITE+REFER** The default setting applies normalization to the **INVITE+REFER+MESSAGE+INFO+OPTIONS+REGISTER+SUBSCRIBE+NOTIFY+PUBLISH** methods.
request-user *setting*	Sets the type of phone number manipulation that the system applies to incoming calls from a server (to the USER field of the Request URI). See User URI normalization for property setting options and descriptions. **Example: set request-user prepend 1978** The default type setting is **no**.

Property name	Description
to-user *setting*	Sets the type of phone number manipulation that the system applies to incoming calls from a server (to the USER field of the To URI). See User URI normalization for property setting options and descriptions. **Example: set to-user strip-off-to 10** The default type setting is **no**.
from-user *setting*	Sets the type of phone number manipulation that the system applies to incoming calls from a server (to the USER field of the From URI). See User URI normalization for property setting options and descriptions. **Example: set from-user thru-registration-plan** The default type setting is **no**.

47. Phone objects

Phones description

The phones configuration object sets AA-SBC to allow replication of phone configuration for supported models. AA-SBC supports the following types of phones:

- Cisco
- Polycomm

➡ **Note:** All but the Polycomm require some level of setup to allow for autoconfiguration. The Polycomm only requires that AA-SBC have a configured TFTP server from which it can access configuration files. See Chapter 70, “TFTP server objects” for information on configuring the TFTP server.

Phones object summary

The following table lists and briefly describes the **phones** objects. See the following chapter for other objects in the CLI hierarchy:

- Chapter 75, “Virtual system partition (VSP) objects”

Object name	Description
`phones`	Opens and enables or disables the phone configuration object.
`  cisco`	Sets autoconfiguration and a password for a Cisco phone.

phones

Purpose

Opens the phones configuration object for editing.

Syntax

```
config vsp phones
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables the phone configuration associated with this AA-SBC device. **Example: set admin enabled** The default setting is **enabled**.

cisco

Purpose

Sets up phone replication and a password for a Cisco phone.

When opening this object, you must specify the model of the phone you are enabling, currently 7960.

Enabling Cisco phone replication

The Cisco phones configuration requires that you enable a TFTP server and configure a location service on AA-SBC. (Enable the TFTP server with the tftp object. See Chapter 38, “Location service objects” for information on configuring location service.)

The basic files necessary for centralized Cisco provisioning must be present in the root TFTP directory (/cxc) prior to configuring the phone. These files are:

- P003-07-5-00.bin
- P003-07-5-00.sbn

- P0S3-07-5-00.bin
- P0S3-07-5-00.loads
- P0S3-07-5-00.sb2
- OS79XX.TXT
- SIPDefault.cnf

Note that the file names may be slightly different, depending on the Cisco software used. For more information on how to configure a Cisco phone for SIP, refer to:

http://www.cisco.com/en/US/tech/tk652/tk701/ technologies_tech_note09186a0080094584.shtml

A phone looks for its specific configuration file on the TFTP server denoted by SIP*macAddress*.cnf when it boots. If it does not find this file, and it does not initially, it looks for SIPDefault.cnf, which contains the basic configuration for any phone. In order for AA-SBC to retrieve the phone's configuration, the 'telnet_level' in this file must be set to 2. The phone then generates its SIP*macAddress*.cnf file and places it on the TFTP server. These files are important, because it is the SIP<MAC>.cnf file that is used for replication.

Phones can be configured using the built in phone interface. Note that the TFTP server you point to from the phone configuration interface must be the TFTP server that was configured on AA-SBC. The outbound proxy must be configured to point to AA-SBC, and 'Register with Proxy' must be set to *yes*.

The phone replication process replicates a phone entirely except for the contact directory. The replication process relies on the primary line—the primary line user is the key that AA-SBC stores in the database. The desired replicant must match the first registration in the phone UI. For example, if, on the parent phone, the Address field for the first registration is jdoe@companyABC.com, another phone can be replicated by entering jdoe@companyABC.com in the first registration. When you press Accept, the phone reboots, registers, and duplicates the parent's configuration.

Syntax

```
config vsp phones cisco phoneType
```

Properties

Property name	Description
password-tag *string*	Specifies the tag associated with the shared secret used to authenticate transactions between the system and this phone. This sets the password for each line and for the phone administrative privileges. See Using passwords and tags for information on the AA-SBC two-part password mechanism. **Example: set password-tag secure** The default setting is **cisco**.

48. Policy objects

The policies configuration object sets both operating policy for enterprise servers and users and denial of service (DOS) policy. This chapter details server and user policy. For information on DOS policy, see Chapter 19, “Denial of Service (DOS) objects”.

Policy configuration description

The policy configuration object allows you to create policies that govern the routing of SIP phone calls and instant messages to recipients, and then back to the original caller or sender. A policy is set of one or more rules, each with defined conditions that operate using a specific SIP session configuration. When a SIP message registers with AA-SBC, the SIP message (such as a SIP INVITE) is processed against the configured policies for matching strings. If a string match occurs in any of the configured policies, then those policies are enforced on that SIP call session.

If there are no policies that match the SIP message and call registration information, the call is either forwarded to the SIP call recipient or the call is dropped based on the settings in the default session configuration.

Rules and condition lists

A policy is identified by a unique name and can contain one or more rules. With each rule, you configure a condition list with a set of properties, as well as the session configuration properties that control the SIP session when the policy is being enforced. Condition list configuration is described in Chapter 15, “Condition list objects”.

Session configuration

The session configuration defines the SIP call session settings to apply to SIP calls for which a configured policy exists. When a SIP call is received at AA-SBC, the system registers the call and checks all policies and rules to determine how the call should be processed, including those services (such as registration, location, authentication, and accounting services) that should be applied to the SIP call. See Chapter 62, "Session configuration objects", for a description of each session configuration object.

For more information on AA-SBC policies, refer to the *Net-Net OS-E – Session Services Configuration Guide*.

Policy configuration object summary

The following table lists and briefly describes the **policy** object. See the following chapter for other objects in the CLI hierarchy:

- Chapter 75, "Virtual system partition (VSP) objects"
- Chapter 15, "Condition list objects"
- Chapter 62, "Session configuration objects"

Object name	Description
`policies`	Opens the policy object, through which you access DOS or session policies.
`  session-policies`	Opens the session policies and sets defaults.
`    policy`	Creates or edits policies that govern the routing of SIP sessions.
`      rule`	Configures rules for the associated policy which can alter the session configuration.
`         condition-list`	Defines the conditions for policy matching. For a description of the condition options, see Chapter 15, "Condition list objects".
`         session-config`	Opens the session-config object for editing. For descriptions of all session configuration subobjects, see Chapter 62, "Session configuration objects".
`  dos-policies`	See Chapter 19, "Denial of Service (DOS) objects".

policies

Purpose

Opens the gateway to the DOS policy and session policy objects.

Syntax

```
config vsp policies
```

Properties

None

session-policies

Purpose

Sets the parameters used to configure policies for this VSP, for enterprise servers, and for users.

Syntax

```
config vsp policies session-policies
```

Properties

Property name	Description
default-policy *policyReference*	Sets the default policy to apply to all SIP sessions. This policy acts as a baseline policy in effect at all times. More specific policy rules may override the default settings. Enter the path to a previously configured policy. **Example: set default-policy vsp policies session-policies policy default** There is no default setting.
outbound-policy	Apply a session configuration policy to a session as it egresses the AA-SBC. Example: set outbound-policy vsp\tls\certificate test

policy

Purpose

Opens the new or existing named policy object for editing.

Syntax

```
config vsp policies session-policies policy name
```

Properties

None

rule

Purpose

Opens the new or existing named policy rule object for editing. Properties of this object enable or disable the policy rule, and set a user-specified description of the rule. From this object you can open the condition-list object to set policy statements or the session-config object to configure baseline SIP session characteristics.

Syntax

```
config vsp policies session-policies policy name rule name
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables the current rule associated with the named policy. If disabled, all other enabled rules under the named policy will still be checked against incoming SIP messages. **Example: set admin enabled** The default setting is **enabled**.
description *text*	Specifies a textual description for this policy rule. Enclose the description in quotation marks and specify up to 64 characters. **Example: set description “Rule to apply to SIP messages from LCS clients belonging to domain company123.com.”** There is no default setting.

session-config

Purpose

Opens the session-config object for editing. It is through this object that you set the session characteristics for SIP calls matching the specified policy. For a description of all session configuration subobjects, see Chapter 62, “Session configuration objects”.

Syntax

```
config vsp policies session-policies policy name rule name
   session-config
```

Properties

None

49. Preferences object

Preferences description

The preferences configuration object allows you to set operational preferences for AA-SBC. The preferences object allows you to add user-defined enumeration strings to the selection of default strings that exist in the AA-SBC configuration file. When editing objects that use enumeration strings (policy predicates, for example), AA-SBC presents both the default strings and those that you have added.

➡ **Note:** Although you can add enumeration strings in the CLI, they are only available through the AA-SBC Management System.

Preferences object summary

The following table lists and briefly describes the **preferences** objects.

Object name	Description
`preferences`	Opens the preferences configuration object.
`Net-Net OS-E-preferences`	Adds enumerations strings to the AA-SBC configuration file.
`summary-preferences`	Opens the summary preferences object to allow configuration of box and cluster status summary page displays.
`cluster-summary-preferences`	Sets the display for the cluster status summary page.
`box-summary-preferences`	Sets the display for the box status summary page.
`monitored-calls-thresholds`	Sets the values that will be highlighted in the AA-SBC Management System Call Logs' Monitored URIs link.

Object name	Description
accounting-calls-preference	Manages the **Accounting Calls** display in the AA-SBC Management System.
select	Sets the fields displayed on the **Accounting Calls** page of the AA-SBC Management System **Call Logs** tab.
tab-order	Manages the tab display in the AA-SBC Management System.
web-tab	Sets the tab order and/or display in the AA-SBC Management System.
click-to-call	Configures click-to-call application profiles.

preferences

Purpose

Opens the preferences configuration object from which you can set specific operations for AA-SBC.

Syntax

```
config preferences
```

Properties

None

Net-Net OS-E-preferences

Purpose

Adds user-defined enumeration strings to the selection of default strings that exist in the AA-SBC configuration file. When editing objects that use enumeration strings (policy predicates, for example), AA-SBC presents both the default strings and those that you have added.

Use this object to add objects configured in other applications, for example in SIP extensions, to the AA-SBC configuration file. Often the extensions contain components that could be useful for defining policy rules. By adding the enumeration strings to the configuration file through this object, the system “remembers” the string so that you can easily use it as a building block in your definitions.

Although you can add strings with this object, the strings will not be available for use from the CLI. They are only available from the GUI. You can confirm that the strings have been added, however, through the **show** object, as shown in the example. In addition, the action of this object occurs automatically in the GUI. If you enter a unique string in one of the categories, the AA-SBC Management System automatically retains that string in the configuration.

Syntax

```
config preferences Net-Net OS-E-preferences
```

Properties

Property name	Description
enum-strings *`type string`*	Adds an entry to the list of displayed entries of the specified type. Type a question mark at the command line to see a list of available types. **Example: set enum-strings timezone homeDST** There is no default setting.
reverse-dns {true | false}	Specifies whether the system should make reverse DNS lookups. **Example: set reverse-dns true** The default setting is **false**.
trap-poll-interval *seconds*	Specifies how often the system checks for new traps. When a trap is detected, it is forward to the trap targets configured with the snmp object. **Example: set trap-poll-interval 15** Enter a value between 1 and 65535; the default setting is **10** seconds.

Property name	Description
phone-path-map *phoneType* {true | false} *URL*	Maps a phone type to the web configuration URL (if it has one). In the Tools section of the AA-SBC Management System, the Phone Registration allows you to click on the IP for a phone and the system dispatches to the vendor phone configuration URL. This property configures the link to the web configuration URL, which is useful in cases where the configuration is not at the top most level. For example, if the phone IP address is 192.168.10.10, the web configuration application may be found at http://192.168.10.10 or it may be at http://192.168.10.10/configuration/. The phone path map allows the administrator to supply an additional path to the web configuration or turn it off completely. For example, if you have a specific Cisco phone of the type CSCO/7, you can add a map entry of type CSCO/7 and then either turn off the configuration altogether or provide an additional path so that the Configure Phone link will go to http://192.168.10.10/configuration for that type of phone. Enter a phone type, whether you want web configuration enabled (set to true if the phone type has management), and the URL for the web configuration page. **Example: set phone-path-map csco/7 true 192.168.10.10/configuration** There is no default setting.
show-unlicensed-features {true | false}	Sets whether the system displays all features or only those licensed for a system. If set to **true**, the system displays all features. For those that are not licensed, the system displays “Available with upgrade.” When set to **false**, the unlicensed option is not displayed. **Example: set show-unlicensed-features false** The default setting is **true**.

Property name	Description
more-than-one-session-in-call {true | false}	*Secondary property.* Sets the system to group sessions in a call based on the call ID. When set to **false**, the default, the system does not group sessions and performance is improved. Set to **true** to configure the call logs sessions search in the AA-SBC to look for more than one session in a call. **Example: set more-than-one-session-in-call true** The default setting is **false**.
channel {none | *name*}	*Secondary property.* Swaps the display of several visual elements within the AA-SBC. By selecting one of the preconfigured channels, you change the AA-SBC display of logo and certain text to reflect the changes implemented for that channel. The default is **none**, which displays the AA-SBC images. Do not change this property unless you have reason to display different channel indicators. **Example: set channel nortel** The default setting is **none**.

Property name	Description
max-config-list-size *integer*	*Secondary property.* Specifies the maximum number of items to display in a configuration list (for example, lists of interfaces or configuration pool entries). When a list contains more items than the number set here, no items of that type are shown in the configuration tree. Instead, they can be displayed on a separate page with paging and search capabilities. The tree contains a link to that page. **Example: set max-config-list-size 50** Enter a value from 10 to 1,000; the default setting is **100** items.
default-call-log-search {enabled | disabled}	*Secondary property.* Controls the initial display that results from clicking the **Call Logs** tab in the AA-SBC Web. If **enabled**, when the tab is clicked the system displays the Sessions page, which lists all sessions in the database for all users. When **disabled**, the database entries are not loaded; the page is blank with the links available on the left. You can then click the **Sessions** link to display the database. **Example: set default-call-log-search disabled** The default setting is **enabled**.

summary-preferences

Purpose

Sets the display page that appears when you either launch the AA-SBC or click on the **Home** tab from within the AA-SBC. You can set the summary display for both box and cluster displays.

Syntax

```
config preferences Net-Net OS-E-preferences summary-preferences
```

Properties

None

cluster-summary-preferences

Purpose

Sets the content of the summary page that displays in the AA-SBC when you choose “Cluster” in the **Get summary for:** pull-down. By default, AA-SBC displays status summary for all choices. When you set **cluster-summary-preference**, AA-SBC overwrites the selection of all status summaries with only the **cluster-summary-preference** you entered. To add additional status summaries, re-execute the command. The order in which you enter the preferences is the order in which they are displayed on the AA-SBC status summary page. If you delete all your entries, AA-SBC returns to the default and displays all cluster status summaries. If your box is not part of a cluster, only the box-addresses and box-summary choices are available.

The following table describes each **cluster-summary-preference** option:

Select...	To display...
box-addresses	the IP address for each configured box in the cluster.
box-summary	the IP address, administrative state, and build information for each configured (physical) box in the cluster.
master-services	the configured host, if applicable, for each master service.
active-vrrp-interfaces	the operational state and configuration of VRRP interfaces, helping to determine interface status to more easily troubleshooting cluster problems.

Syntax

```
config preferences Net-Net OS-E-preferences summary-preferences
   cluster-summary-preferences {box-addresses | box-summary |
   master-services | active-vrrp-interfaces}
```

box-summary-preferences

Purpose

Sets the content of status summary page that displays in the AA-SBC when you choose “Box” in the **Get summary for** pull-down. By default, AA-SBC displays status summary for all choices. When you set **box-summary-preference**, AA-SBC overwrites the selection of all status summaries with only the **box-summary-preference** you entered. To add additional status summaries, re-execute the command. The order in which you enter the preferences is the order in which they are displayed on the AA-SBC status summary page. If you delete all your entries, AA-SBC returns to the default and displays all box status summaries.

The following table describes each **box-summary-preference** option:

Select...	To display...
box-status	the IP address, administrative state, and build information for the box. Links on the page provide access to interface, process, and sensor information.
master-services	the list of master services hosted on the selected box.
up-time	system uptime since the last reboot.
system-info	CPU utilization statistics over the time frame specified for that status provider. Links on the page provide access to memory and alert data.
registration-info	the total number of client bindings stored in the registration table, broken down into ignored, terminated, and declined registrations. A link on the page displays the total number of entries in the location cache.
call-info	the call counts of active, dropped, and total calls, as well as a link to display call trends.
call-duration	the average and maximum call duration, as well as a link for a session duration trend graph.
location-info	a count of entries in the location cache and location bindings table.

Syntax

```
config preferences Net-Net OS-E-preferences summary-preferences
   box-summary-preferences {box-status | master-services | up-time |
   system-info | registration-info | call-info | call-duration |
   location-info}
```

monitored-calls-thresholds

Purpose

Sets the values that will cause the display in the AA-SBC Web Call Logs' **Monitored URIs** link to display in red, highlighting that call performance has crossed the set threshold. The **Monitored URIs** link executes and then displays the results of the QoS loopback monitoring tests.

Syntax

```
config preferences Net-Net OS-E-preferences monitored-calls-thresholds
```

Properties

Property name	Description
mos-threshold `threshold`	Sets the value below which the Mean Opinion Score (MOS) result triggers notification. MOS is a subjective measurement and an “opinion” of the audio quality heard by the listener on a phone. The MOS measurement reveals the call quality from 1 (pure noise) to 5 (pure fidelity). **Example: set mos-threshold 4.2** The default setting is **4.0**.
latency-threshold *milliseconds*	Sets the value above which the result of the QoS loopback monitoring test triggers notification. Latency is defined as the delay in getting RTP packets from the system to the endpoint (e.g., phone). **Example: set latency-threshold 500** The default setting is **400** milliseconds.

Property name	Description
jitter-threshold *milliseconds*	Sets the highest allowable packet variation (jitter) allowed on a call. When the jitter exceeds this configured threshold, the system highlights the results. **Example: set jitter-threshold 45** The default setting is **60** milliseconds.
packets-dropped-percent-threshold *percent*	Sets the maximum percentage of dropped packets allowed. When the percentage of packets dropped exceeds the configured threshold, the system highlights the results. The number of packets sent is determined by the type and duration set in the **loopback** action. **Example: set packets-dropped-percent-threshold 15** The default setting is **20** percent.

accounting-calls-preference

Purpose

Manages the display of the call detail record fields in the AA-SBC Web. See the *Net-Net OS-E – Session Services Configuration Guide* for a description of each field.

Syntax

```
config preferences Net-Net OS-E-preferences
   accounting-calls-preference
```

Properties

None

select

Purpose

Sets the fields to display on the **Accounting Calls** page of the AA-SBC Web **Call Logs** page. By default, fields are displayed in the order in which they are entered. To change the order, use the CLI move command or the arrows within the AA-SBC Web listing. If you do not configure this property, the AA-SBC displays Setup Time, Type, Method, From, To, Call ID, and the calculated MoS score on the source leg. To view all accounting fields for a call, click the Call Record link for the call on the **Accounting Calls** page.

To configure this object, re-execute the object for each column you want displayed.

Syntax

```
config preferences Net-Net OS-E-preferences
   accounting-calls-thresholds select columnName
```

Properties

None

tab-order

Purpose

Manages the display and/or order of tabs in the AA-SBC Web. See the *Net-Net OS-E – Using the NNOS-E Management Tools* for a description of each tab.

Syntax

```
config preferences Net-Net OS-E-preferences tab-order
```

Properties

None

web-tab

Purpose

Sets the tabs to display in the AA-SBC Web. By default, and with full permissions, the following tabs are displayed:

- Home
- Configuration
- Status
- CallLogs
- EvenLogs
- Actions
- Service
- Keys
- Access
- Tools
- Portal

If you do not see all of these tabs, it is probably because access to that capability is turned off.

Note that you must re-execute this object for each tab you want displayed. For example, if you execute only once, supplying **Actions** as the *tabName*, the **Actions** tab will be the only tab visible after you save the configuration. To re-order the tabs put continue to display them all, you must execute the object 11 times.

Syntax

```
config preferences Net-Net OS-E-preferences tab-order web-tab tabName
```

Properties

None

click-to-call

Purpose

Configures click-to-call application profiles. These profiles are then referenced with the web-service **application** property. The click-to-call function is reserved for demonstration use only.

Syntax

```
config preferences click-to-call
```

Properties

Property name	Description
record {enabled | disabled}	Specifies whether to record the transaction. If **disabled**, the system has a record of the call but does not retain the content. The system applies the session configuration referenced in the **dont-record-entry** property. When **enabled**, the system records the call, which you can then view through the AA-SBC **Call Logs**, and applies the session configuration referenced in the **record-entry** property. **Example: set admin disabled** The default setting is **enabled**.
phone-to-call *AOR*	Specifies the address of record to which the system is dialing out. **Example: set phone-to-call sip:5555555432@192.168.10.10** There is no default setting.
record-entry *sessConfigReference*	References a session configuration to apply if the **record** property is set to **enabled**. **Example: set record-entry “vsp session-config-pool-entry entry click-to-record”** There is no default setting.
dont-record-entry *sessConfigReference*	References a session configuration to apply if the **record** property is set to **disabled**. **Example: set dont-record-entry “vsp session-config-pool-entry entry dont-record-click”** There is no default setting.

50. Presence database objects

Presence database description

The presence database is a tool to save state information about presence “watchers” and the *presentities* they subscribe to, thereby allowing different *presence domains* to interact. (A presentity, according to RFC 2778, *A Model for Presence and Instant Messaging*, is “an entity of interest to a presence service.”) Presence domains can be federated to allow for sharing of presence information. For example, a Sametime community connected to an LCS community that share presence via a AA-SBC device would have two presence domains. The presence database compensates for differences in presence-sharing protocol use, providing a vendor-neutral repository of subscriber information. It also provides AA-SBC with the data necessary to make intelligent call-forking decisions.

AA-SBC populates the presence database in two cases:

- when different presence domains are connected (federated) through AA-SBC (for example, Sametime-to-LCS). Like configurations do not add entries in the database.
- when AA-SBC sees REGISTER messages from Windows Messenger clients, indicating whether a particular client is active or not. (This is only applicable in a topology where AA-SBC sits between clients and the LCS server.)

The presence database stores information such as the time a watcher has subscribed to a presentity, the expiration interval, and the presentity state. When configured, AA-SBC adjusts the online state if it has seen no-resubscribes. AA-SBC does not use the database to control the presence of any IM clients. Instead, the presence database information supports bridging different “flavors” of IM and advising when and where to fork a call.

Each top-level watcher entry in the presence database stays for the life of the VSP. The subscribed-to list grows and shrinks as subscribers are added and removed. When the database feature is enabled, AA-SBC cleans the database entries based on a wider range of information, making presence information more accurate.

Presence database object summary

The following table lists and briefly describes the presence-database object. See the following chapter for other objects in the CLI hierarchy:

- Chapter 75, “Virtual system partition (VSP) objects”

Object name	Description
`presence-database`	Enables or disables the presence database scanning feature and sets the scan interval.

presence-database

Purpose

Enables and sets the interval for the AA-SBC presence database scan. AA-SBC always builds the presence database; this object sets the mechanism for cleaning and updating it. Enabling the database scan feature causes additional overhead but increases the accuracy of call forking. It is most useful when you have a database of significant size.

Note that when you are setting values for maximum watchers and presentities, these numbers only apply to users outside of your domain. Users within the domain do not count against the limits that you set.

To clear the presence database, use the presence **flush** action. To view a list of watchers, use **show presence-cache**; to view remote entities, use **show presence-presentities-external**.

Syntax

```
config vsp presence-database
```

Properties

Property name	Description
database-scan {enabled | disabled}	Enables or disables the presence database scan feature. If **enabled**, the system scans and cleans the database based on SIP traffic (SUBSCRIBE, NOTIFY, and REGISTER requests) and expiration interval. This allows an entry with a status of offline or unknown, for a reason other than unsubscribing (e.g., unplugging a laptop without shutting down), to be removed. If **disabled**, the system cleans the database based on SUBSCRIBE requests only. This improves performance but may result in slightly less efficient call forking when the system still “believes” that a user is online. **Example: set database-scan enabled** The default setting is **disabled**.
database-scan-interval *seconds*	Sets the interval at which the system scans the presence database. Setting the interval too low can have a noticeable negative impact on performance. **Example: set database-scan-interval 7200** Enter a value from 30 to 86,400. The default setting is **3600** seconds.
max-watcher *value*	Specifies the maximum number presence watchers allowed in the database. This value, combined with the dbMaxPresentities value, determines allowable database size. (Note that the total size is ultimately limited by your license.) **Example: set max-watcher 6000** Enter a value within the allowable size of your license; the default setting is **32768**.
max-presentities *value*	Specifies the maximum number presentities (those remote entities being watched) allowed in the database. This value, combined with the dbMaxWatchers value, determines allowable database size. (Note that the total size is ultimately limited by your license.) **Example: set max-presentities 50000** Enter a value within the allowable size of your license; the default setting is **262144**.

Property name	Description
max-pres-per-watcher *value*	Sets a limit on the number of presentities to which a watcher can subscribe. In a federated IM environment, this number should match (approximately) the number of active contacts from other domains. Note that when this value is reached, the system does not disallow the traffic. It simply does not add the information to the presence database. This feature is particularly important for phone presence because typically phones do not explicitly unsubscribe from a presentity. To compensate for this, the system does a partial cleaning without the overhead of a full pruning. When a phone subscribes to a presentity but the limit has been reached, the system erases any entries that have expired. **Example: set max-pres-per-watcher 32** The default setting is **16**.
non-watcher-entries {enabled | disabled}	Specifies whether to allow non-watchers entries into the database. In the default form (**disabled**), entities that have not issued SUBSCRIBE messages are not shown as watchers and the system will not retain any presence information about these. When **enabled**, the system adds these entities as watchers based on presence information seen in NOTIFY messages. Note, however, that the presentity lists for these will be empty as they are not true watchers (i.e., they have never issued SUBSCRIBEs). **Example: set non-watcher-entries enabled** The default setting is **disabled**.

Property name	Description
lcs-append-sub-response-pidf {enabled | disabled}	Specifies whether to append offline Presence Information Data Format (PIDF) to the first SUBSCRIBE response to an LCS server if the Sametime presentity is offline or unknown. Use this property when your configuration includes a ST-to-LCS federation. When enabled, the system appends a presence information document on a 200/ OK responses onto a SUBSCRIBE message sent from LCS to Sametime if the system knows the current state of the Sametime presentity. Appending the PIDF allows Office Communicator clients to correctly indicate a status of ‘Offline’ instead of ‘Unknown' when Sametime contacts are not available. When disabled, the system does not modify responses to SUBSCRIBE messages sent from LCS to Sametime. **Example: set lcs-append-sub-response-pidf disabled** The default setting is **enabled**.
lcs-subscribe-interval *seconds*	Sets the interval at which LCS clients resubscribe to a Sametime domain. (This setting compensates for issues with the Sametime SIP Connector.) The large default value allows Windows Messenger and Office Communicator clients to subscribe with an entry that last seven days, minimizing the amount of cross-domain resubscription traffic. **Example: lcs-subscribe-interval 907200** The default setting is **604800**.

51. Pre-session configuration objects

Pre-session configuration description

The pre-session-configuration object allows you to globally apply SIP settings to your network before SIP call sessions are established. SIP methods that you want to shield from the network, for example, can be blocked using settings in the **pre-session-configuration** object. This means that you do not need to create a policy rule to block a particular SIP method that is globally forbidden from your network.

Pre-session-config object summary

The following table lists and briefly describes the **pre-session-config** object. See the following chapter for other objects in the CLI hierarchy:

- Chapter 75, "Virtual system partition (VSP) objects"

Object name	Description
`pre-session-config`	Opens the object through which you set the parameters to alter SIP traffic before a session is established.
`block-method-settings`	Specifies the SIP method(s) to block on incoming traffic.
`sip-header-settings`	Enables or disables the configured rules used to block traffic prior to session establishment.
`rule`	Defines the action to take if AA-SBC finds a match in the predicate statement contained in the condition list.

pre-session-config

Purpose

Opens the pre-session-config object for editing. Through this object you set the parameters used by the VSP to alter SIP traffic before a session is established.

Syntax

```
config vsp pre-session-config
```

Properties

Property name	Description
unregistered-sender-directive {allow | discard | refuse [*resultCode*] [*resultString*]}	Sets the action the system takes when it receives a packet with an unknown sender in the “From” field of the INVITE packet. Use the **registration-requirement-level** setting in the route or source-route object to define what is considered unknown. Select one of the following actions: • **allow**—the system permits the packet to proceed toward its destination. • **discard**—the system immediately discards the packet. • **refuse**—the system discards the packet but sends a response to indicate having done so. The response includes an error code (default of 400 but you can enter any value between 400 and 699) and an optional description. **Example: set unregistered-sender-directive refuse 404 “unknown sender”** The default setting is **allow**. If you select **refuse**, the default result code is 400.
optional-header-error-handling {strip | ignore | discard | reject [*resultCode*] [*resultString*]}	Determines how the system handles optional header parsing. Select one of the following actions: • **strip**—the system strips the malformed header in its entirety. • **ignore**—the system ignores the error and sends the malformed header on the egress leg. • **discard**—the system immediately discards the packet. • **reject**—the system discards the packet but sends a response to indicate having done so. The response includes an error code (default of 400 but you can enter any value between 400 and 699) and an optional description. **Example: set optional-header-error-handling ignore** The default setting is **strip**. If you select **reject**, the default result code is 400 with the string “Bad Request - Optional Header Failed Parsing.”

block-method-settings

Purpose

Enables or disables blocking of SIP methods and specifies the SIP method to block. When a method type is blocked, AA-SBC drops the packet. Repeat the command to block additional SIP methods.

The following table lists the methods that AA-SBC can block, along with brief definitions from the related RFC:

Method	Description	Found in...
INVITE	Asks a server to establish a session.	RFC 3261, *SIP: Session Initiation Protocol*
ACK	Facilitates reliable message exchange for INVITEs.	RFC 3261
OPTIONS	Allows a UA to query another UA or a proxy server as to its capabilities.	RFC 3261
BYE	Terminates a specific session or attempted session.	RFC 3261
CANCEL	Asks the UAS to cease processing the request and to generate an error response to that request.	RFC 3261
REGISTER	Sends a request to a... registrar. A registrar acts as the front end to the location service for a domain, reading and writing mappings based on the contents of REGISTER requests.	RFC 3261
MESSAGE	Allows the transfer of Instant Messages.	RFC 3428, *Session Initiation Protocol (SIP) Extension for Instant Messaging*
INFO	Allow for the carrying of session related control information that is generated during a session.	RFC 2976, *The SIP INFO Method*

Method	Description	Found in...
NOTIFY	Contains the modified session description.	RFC 2848, *The PINT Service Protocol: Extensions to SIP and SDP for IP Access to Telephone Call Services*
SUBSCRIBE	Indicates that a user wishes to receive information about the status of a service session.	RFC 2848
REFER	Requests that the recipient REFER to a resource provided in the request...and provides a mechanism allowing the party sending the REFER to be notified of the outcome of the referenced request.	RFC 3515, *The Session Initiation Protocol (SIP) Refer Method*
PRACK	Plays the same role as ACK, but for provisional responses.	RFC 3262, *Reliability of Provisional Responses in the Session Initiation Protocol (SIP)*
PUBLISH	Provides a framework for the publication of event state information.	RFC 3903, *Session Initiation Protocol (SIP) Extension for Event State Publication*

Syntax

```
config vsp pre-session-config block-method-settings
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables the blocking of SIP methods before SIP sessions are established. **Example: set admin enabled** The default setting is **enabled**.
block-method *methodType*	Specifies the SIP method(s) to block from the network. Re-execute the command to add each block method. See the above table for method descriptions. **Example: set block-method refer** There is no default setting.

sip-header-settings

Purpose

Enables or disables the SIP header rules (set with the rule object) that are applied to the network before a SIP session is established.

Syntax

```
config vsp pre-session-config sip-header-settings
```

Properties

Property name	Description
`admin {enabled | disabled}`	Enables or disables the SIP header policy that is applied to the network before SIP session establishment. You configure the rules of the policy with the rule object. **Example:** set admin enabled The default setting is **enabled**.

rule

Purpose

Sets an optional rule description, the conditions of the rule, and the type of action to apply to SIP headers that match those conditions. To open the rule object, specify a name. Initially, the order in which they were created establishes the precedence for the rule (if you create multiple rules). Use the **move** command to change the order.

A condition is a predicate statement that AA-SBC matches the SIP headers against. If a header matches any of these statements, AA-SBC takes the action defined by the **action** property. Note that the conditions are AND’d together.

Follow these rules when creating conditions:

- If you enter a header name only, AA-SBC applies the **action** to that header.
- You can enter only one header name. To match on more than one, create multiple rules.
- If you enter a header value only, AA-SBC applies the action if any header matches that value.
- You can enter more than one value, but should do so with extreme care, as the rule will take a single action against all matches.
- If you enter a name and value, AA-SBC applies the action to the named header if it has the specified value.

Syntax

```
config vsp pre-session-config sip-header-settings rule name
```

Properties

Property name	Description
description *text*	Sets the user-specified text description for the rule. Use the **show -v** command from the **sip-header-settings** level to see all configured rules with descriptions. **Example: set description “SIP header policy to apply prior to session establishment.”** There is no default setting.

Property name	Description
condition {match-header *headerName* | match-header-and-value *headerName regExp* | match-value *regExp*}	Sets whether to match on a name of a SIP header and/or a value in one of the fields. (See the more detailed explanation in this command description.) Select one or more of the following to set the trigger for the action: • **match-header**—sets the name of the header to match on. To see possible name matches, enter **set condition match-header ?** at the prompt. • **match-header-and-value**—sets the header name and field to match against. In this case, both entries must match. • **match-value**—matches all SIP headers against the text string you enter. Enclose a string with spaces within quotation marks. **Example:** `set condition match-header To` `set condition match-header-and-value From bob` There is no default setting.
action {discard-packet | strip-header | alter-header *newSipHeader*}	Sets the action to apply to packets in which the conditions of this rule are met. Select one of the following: • **discard-packet**—the system immediately discards the packet. • **strip-header**—the system removes the SIP header from the packet. Use this, for example, if a particular header causes problems for another SIP device in the network. • **alter-header**—the system changes the content of the header to the text you supply. If your condition list contained a match-name statement, the system alters the named header. If your condition list contained only a value, the system alters all headers that contain that value. **Example: set action alter-header 800** There is no default setting.

Pre-session configuration objects

52. Processes objects

Processes description

The **processes** objects allow you to configure memory allocation settings on a per-process basis. The settings in these objects should not be modified unless specifically instructed to do so by Technical Support personnel.

When the system boots, AA-SBC tallies the amount of physical memory in the system and scales each processes heap based on that total.

Processes object summary

The following table lists and briefly describes the **processes** object. See the following chapter for other objects in the CLI hierarchy:

- Chapter 14, “Cluster, box, and interface objects”

Object name	Description
`processes`	Opens the processes object from where you set per-process configuration.
`process`	Sets various memory allocation settings for a specified process.

processes

Purpose

Opens the processes object, from where you specify memory configuration settings for each process.

Syntax

```
config box processes
config cluster box number processes
```

Properties

None

process

Purpose

Sets memory allocations for each specified process. When opening this object, specify the name of the process for which you would like to modify the parameters. Enter a process type to open this object. Type a question mark at the command line to display possible process types.

Note: Do not change these settings without explicit instructions to do so from Technical Support personnel.

The **process** configuration settings are dynamic. If AA-SBC runs out of memory in one of the heaps, you can change the configuration and AA-SBC will expand the heap. If you make a value smaller than the current heap size, AA-SBC will not free any memory, but future allocation attempts will fail.

When setting the heap sizes, you select either the default value, a maximum, or a specific value. You can display the current values with the **show system-heap** command.

These options are defined as follows:

Setting	Definition
default	AA-SBC calculates default values at boot time, based on the total amount of memory available to the system.
max	There is no limit to the heap size; AA-SBC is free to use all the memory available on the process.
value	AA-SBC limits the heap size to the specific value you enter. Enter the keyword **value** and the number of megabytes to assign.

Syntax

```
config box processes process processName
config cluster box number processes process processName
```

Properties

Property name	Description
system-heap-init {default | max | value *megabytes*}	Specifies the initial amount of memory that the system reserves for a process. **Example: set system-heap-init value 2048** The default setting is **default**. If you select **value**, enter a number of megabytes between 8 and 3072.
system-heap-max {default | max | value *megabytes*}	Specifies the total amount of memory that the system can use on a process. **Example: set system-heap-max 2048** The default setting is **default**. If you select **value**, enter a number of megabytes between 8 and 3072.
system-heap-locked {disabled | enabled *megabytes*}	*Do not modify this value unless told to do so by Technical Support.* Specifies the amount of memory that should be locked. A value of 0 configures the system to lock no memory. Locked memory cannot be swapped out by the Linux kernel. **Example: set system-heap-locked enabled 128** The default setting is **disabled**.

Property name	Description
system-heap-shrink-interval *minutes*	Specifies how often the system checks the system heap to see if any memory can be returned to the operating system. A value of 0, the default, specifies that the system not make that check. **Example: set system-heap-shrink-interval 90** Enter a value between 0 and 1440. The default setting is 60 minutes.
pools-shrink-interval *minutes*	Specifies how often the system checks its memory pools to see if any memory can be returned to the system heap. A value of 0, the default, specifies that the system not make that check. **Example: set *pools*-shrink-interval 20** Enter a value between 0 and 1440. The default setting is 10 minutes.
tls-heap-max {default | max | value *megabytes*}	Specifies the portion of the **system-heap-max** that is used for TLS processing. If set to **default**, the maximum TLS heap size is calculated based on the value of the **max-number-of-tls** property in the vsp configuration. **Example: set tls-heap-max 512** The default setting is **default**. If you select **value**, enter a number of megabytes between 8 and 3072.

53. Proxy interface object

Proxy description

Generic proxying is the mechanism AA-SBC uses to allow SIP clients who are behind firewalls or NATs to reach each others via AA-SBC. Because SIP clients can use protocols other than SIP for getting their initialization and configuration information, setting up web conferences, and so on, SIP clients can use the AA-SBC generic proxy to forward those protocols.

Typical applications include:

- Service providers providing a single IP address/domain name for subscribers, whether for SIP, FTP, or HTTP. (Many phones use HTTP to obtain their configuration files, or to download XML-based scripts that control their LCD displays and button assignments.)
- Enterprises using AA-SBC as an edge proxy for OCS 2007 traffic. OCS 2007 adds web conferencing to IM, presence, and SIP-based voice and video. Web conferencing uses HTTPS and a protocol called PSOM (Persistent Shared Object Model).

Proxy object summary

The following table lists and briefly describes the **proxy** object. See the following chapters for other objects in the CLI hierarchy:

- Chapter 14, “Cluster, box, and interface objects”
- Chapter 77, “VLAN objects”
- Chapter 35, “IP objects”

Object name	Description
`proxy`	Configures and enables generic proxying for an interface.

proxy

Purpose

Configures and enables a generic proxy interface, allowing non-protocol aware communications via AA-SBC. Set the proxy for each transport protocol and port combination used to reach a destination.

Syntax

```
config cluster box number interface ethX ip name proxy {tcp | udp} port
config cluster box number interface ethX vlan number ip name proxy {tcp
    | udp} port
config box interface ethX ip name proxy {tcp | udp} port
config box interface ethX vlan number ip name proxy {tcp | udp} port
```

Properties

Property name	Description
admin {enabled | disabled}	Sets the administrative state of the proxy configuration. When **disabled**, the parameters of generic proxy interfaces can still be configured, but proxying does not occur. **Example: set admin disabled** The default setting is **enabled**.
destination *ipAddress*	Specifies the IP address of the internal destination (the address behind the private firewall). **Example: set destination 172.24.3.99** There is no default destination.
ports *number*	Specifies the number of ports the system allocates to the proxy interface. **Example: set ports 500** The default number of ports is **1000**.

54. RADIUS-group objects

RADIUS group description

A RADIUS group is a uniquely named object that defines the authentication and accounting services associated with a group of RADIUS servers. Including a RADIUS group in the VSP configuration allows the AA-SBC system (the RADIUS client) to perform user authentication and forward accounting and SIP call detail records to the RADIUS servers.This means that you have flexibility to create as many unique RADIUS groups as you need, and include them with the VSPs of your choice.

Setting server priority

AA-SBC allows you to set server priority to influence which server receives authentication requests. To use this feature, set the **authentication-mode** property in the radius-group object to **prioritized**. Set the priority for the server with the **priority** property of the server object. AA-SBC then manages authentication requests using the following logic:

1. AA-SBC always sends an authentication request to the server with the highest priority. The lower the number, the higher the priority.
2. If the request times out, AA-SBC sends the request to the next-highest-priority server. This timeout status is applicable for that request only. AA-SBC will forward the next request to the highest priority server.
3. AA-SBC continues with this action until either a server replies with an Accept or a Reject, or until there are no more configured servers. If there are no more servers to try, AA-SBC rejects the call.

Note that in **prioritized** mode, AA-SBC does not determine that servers are dead due to consecutive failures. As long as a server is enabled in the configuration, AA-SBC continues to forward requests, regardless of the number of failures.

When configuring for prioritization, it is important to set different priority values for the servers. Otherwise, AA-SBC will randomly select from servers with the same value, negating the effects of prioritized mode. If that should happen, AA-SBC will generate an event indicating that multiple servers have the same priority. The following two example illustrate how AA-SBC forwards requests with multiple servers of the same priority:

- Server A has a priority of 1, and servers B and C have a priority of 2. AA-SBC sends all requests to server A, with the highest priority, first. If A does not respond, AA-SBC picks randomly between B and C.
- Servers A and B have a priority of 1, and server C has a priority of 2. AA-SBC selects randomly between A and B, and sends all requests to that server first. If that server times out, AA-SBC sends all requests to the other highest-priority server. (For example, if AA-SBC picks A first, and it times out, it then sends requests to B, not C.)

RADIUS-group object summary

The following table lists and briefly describes the **radius-group** objects. See the following chapter for other objects in the CLI hierarchy:

- Chapter 75, “Virtual system partition (VSP) objects”.

Object name	Description
`radius-group`	Opens or creates the specified RADIUS accounting group configuration object on the AA-SBC device.
`server`	Opens the RADIUS server configuration object on AA-SBC based on the specified number.
`call-field-filter`	Specifies fields from the CDR to be written to the database target. See the command description in Chapter 6, “Accounting objects”.

radius-group

Purpose

Configures a RADIUS group, to which you add servers using the server object. Setting up a RADIUS group in one or more VSP configurations allows the AA-SBC system (the RADIUS client) to perform SIP traffic authentication and to forward accounting and SIP call detail records to the RADIUS servers. (To setup authentication for user access, use the access radius object.)

Specify the new or existing RADIUS group name using up to 16 alphanumeric characters with no blank spaces.

Syntax

```
config vsp radius-group targetName
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables the system RADIUS server group configuration. When **enabled**, the system forwards SIP call detail records to configured RADIUS group server(s). **Example:** **`set admin enabled`** The default setting is **enabled**.
accounting-mode {duplicate | round-robin| fail-over retries}	Sets the RADIUS group accounting operational algorithm: round-robin, failover, or duplicate. • **round-robin**—If you configure multiple accounting servers in the accounting group, the round robin algorithm performs continued accounting requests to primary and secondary servers until a valid accounting response is received. • **duplicate**—The duplicate algorithm issues multiple duplicate accounting requests to all servers in the RADIUS accounting group. A duplicate accounting request uses the same client source IP address and source UDP port. • **fail-over**—If you configure multiple accounting servers, the failover algorithm forwards accounting requests to secondary servers should the current accounting server fail. You can specify up to 256 failover servers. **Example:** **`set accounting-mode round-robin`** The default setting is **duplicate**.

Property name	Description
authentication-mode {round-robin| fail-over retries | prioritized}	Sets the RADIUS group authentication operational algorithm. If you configure multiple authentication servers in the RADIUS group, select either: • **round-robin**—the round robin algorithm performs continued authentication requests to primary and secondary servers until a valid authentication response is received. • **fail-over**—the failover algorithm forwards authentication requests to secondary servers should the current authentication server fail. You can specify up to 256 failover attempts to other servers. • **prioritized**—the system forwards authentication requests to the server with the highest assigned priority. If that server does not respond, the system forwards the request to the next highest priority server. Set the priority with the **server priority** property. See Setting server priority for more information. **Example: `set authentication-mode round-robin`** The default setting is **fail-over** with **3** retries.
type *type*	Sets the type of SIP accounting record to use. Currently, the only valid SIP accounting record type is Cisco. **Example: `set type cisco`** The default setting is **cisco**.
included-in-default {true | false}	Specifies if this RADIUS group is to be included in the default RADIUS authentication and accounting target group. If set to **true**, authentication and accounting requests are forwarded to this group if there are no configured policies that govern or redirect RADIUS requests to other servers. **Example: `set included-in-default true`** The default setting is **true**.

Property name	Description
digest-attributes-format {draft-sterman-aaa-sip-03 | draft-ietf-radext-digest-auth-05 | rfc-4590}	Sets the correct Digest authentication attributes format for use with RADIUS. Select either: • **draft-sterman-aaa-sip-03**—set to this experimental format if you are using FreeRADIUS. • **draft-ietf-radext-digest-auth-05**—set to this early proposed standard if you are using Steel-Belted RADIUS. • **rfc-4590**—set to RFC 4590 is you are using the standard RADIUS. **Example: `set digest-attributes-format rfc-4590`** The default setting is **draft-sterman-aaa-sip-03**.
send-session-id {true | false}	Specifies whether the system correlates RADIUS access requests with accounting requests. When **true**, the system sends the Acct-Session-ID attribute in its RADIUS auth-requests. When **false**, this attribute is sent only in accounting messages. **Example: `set` send-session-id `false`** The default setting is **true**.
include-digest-domain-in-user-name {enabled | disabled}	Specifies whether to append the user's domain name to the RADIUS User-Name attribute. Enable this property if the RADIUS server requires the domain name to be included in the attribute. If the User-Name attribute already contains a domain name, the system does not take any action. **Example: set include-digest-domain-in-user-name enabled** The default setting is **disabled**.
send-user-agent {true | false}	Specifies whether to include the User-Agent header value in the RADIUS Auth-Request message. If set to **true**, AA-SBC includes the User-Agent header in the Connect-Info RADIUS attribute. **Example: `set` send-user-agent `true`** The default setting is **false**.

Property name	Description
service-type *method serviceType*	Maps a RADIUS service type to a SIP message type. If the system authenticates a message type that has a mapped service type, it will include that Service-Type attribute in the RADIUS request. If a service type has not been mapped to the message type the system is authenticating, but there is a mapping for the message type OTHER, the system includes the OTHER service type in the request. If there is no mapping for the actual or the OTHER method, then the system does not include any Service-Type attribute in the request. **Example: set include-digest-domain-in-user-name enabled** The default setting is **disabled**.
application	Enter the RADIUS application ID for the servers in this group. The following are valid options: • authentication—use SIP authentication • routing—use SIP routing Note that Java accounting ignores this setting and considers all RADIUS servers as candidates for RADIUS accounting. Example: set application routing The default setting is **authentication**.

server

Purpose

Identifies and configures the RADIUS servers that are part of this RADIUS group. Enter a host name or IP address to identify the server.

Syntax

```
config vsp radius-group name server serverName
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables the system RADIUS authentication and accounting server configuration. When enabled, authentication and SIP call accounting records are forwarded to the specified server IP address and port numbers. **Example: `set admin enabled`** The default setting is **enabled**.
authentication-port *portNumber*	Sets the UDP port number over which the system RADIUS client sends authentication requests to the RADIUS server. **Example: `set authentication-port 1800`** Enter a value in the range of 1 to 65535; the default setting is UDP port **1812**.
authentication-sockets *sockets*	Sets the number of sockets reserved for request IDs on a server. With one socket, the default, the 8-bit number space allows up to 255 outstanding requests per server. Assign additional sockets if you have a high-volume application that requires sending many requests at one time. Each additional socket increases capacity by 255 requests. **Example: `set authentication-sockets 4`** Enter a value in the range of 1 to 8; the default setting is **1** socket.
accounting-port *portNumber*	Sets the UDP port number over which the system RADIUS client sends accounting requests to the RADIUS server. **Example: `set accounting-port 1801`** Enter a value in the range of 1 to 65535; the default setting is UDP port **1813**.

Property name	Description
secret-tag *text*	Specifies the shared secret used to authenticate transactions between the system RADIUS client and the RADIUS server. See Using passwords and tags for information on the AA-SBC two-part password mechanism. **Example:** **`set secret-tag abc123xyz`** Enter up to 32 alphanumeric characters. There is no default setting.
timeout *milliseconds*	Specifies the time (in milliseconds) to elapse before an accounting or authentication request to a RADIUS server times out. If the request times out, the system retries the request for the specified number of attempts before the request is forwarded to the next RADIUS server in the configuration. **Example:** **`set timeout 1500`** Enter a value from 1 to 65535; the default setting is **1000** milliseconds (1 second).
retries *integer*	Sets the number of times the system retransmits an accounting or authentication request if the RADIUS server does not respond. **Example:** **`set retries 5`** Enter a value from 2 to 5; the default setting is **3** attempts.

Property name	Description
window *integer*	Sets the maximum number of simultaneous requests the system client can send to the RADIUS server. Note that if you set multiple sockets with the **authentication-socket** property, this window value is a per-socket allowance. **Example: set window 255** Enter a value from 8 to 255; the default setting is **64** requests.
priority *integer*	Configures a priority for the server. Set this property if the **authentication-mode** property of the radius-group object is set to **prioritized**. The lower the value, the higher the priority. Note that each server in a RADIUS group must have a different priority for prioritization to work correctly. See Setting server priority for more information. **Example: set priority 2** Enter a value between 1 and 99. The default setting is **1**.

55. Registration plan objects

Registration plan description

The registration plan allows you to delegate, proxy, forward, or redirect a SIP requests to a particular gateway based on the dial prefix or domain suffix. (The registration-plan is used when AA-SBC is at the non-originating end of a call.) By default, registration plans apply to the following types of requests:

- REGISTER
- SUBSCRIBE
- NOTIFY
- PUBLISH

Note that you can change whether a message type uses the registration plan using the **sip-message-plan** property of the **settings** object. (You cannot change the plan type for REGISTER messages, they always use a registration plan.)

When you configure components within the registration plan, AA-SBC adds those entries to the corresponding registration plan table. AA-SBC does lookups on the tables to find matching criteria that define further selection and/or alteration. The components are considered in order of processing:

1. **normalization**
2. **arbiter**
3. **route**

In addition, all plans with an associated condition-list appear in the table before those without. Otherwise, AA-SBC determines the table order based on the match statement type and value. See Pattern Match Precedence to determine precedence of the match types.

Registration plans use a condition-list as a “first pass” filter when matching a plan entry. When AA-SBC receives an incoming request of the type listed above, it compares the request against the registration plan table entries with configured conditions, and returns a list of matching plans. The match statements within the plan components then determine the next level of filtering. If there are still multiple matches, the **priority** setting within the component determines the order of selection.

Note: Condition lists are common to several objects and are documented in Chapter 15, “Condition List Objects.”

AA-SBC determines which action to take using either the **action** property of the **route** object or the by matching on the criteria set in the **proxy** or **source-proxy** objects. When AA-SBC **delegates** a REGISTER, it changes the contact to its own address. This allows an upstream registrar to forward subsequent messages to AA-SBC rather than to the phone client directly. (Be sure to specify a peer if the action is set to **delegate**.) If the registration-plan specifies to **forward** a REGISTER, the system forwards the REGISTER unchanged to the upstream registrar. An action of **redirect** causes the system to respond with alternative registrars’ contact information and the instructions to resend the REGISTER to an alternative registrar as specified.

The configuration of the registration plan determines the entries in the registration routing table. It is this table that determines AA-SBC lookup behavior. If a server that is referenced in a registration plan becomes unavailable, the system removes the entry from the registration routing table. However, the entry remains in the configuration.

The registration routing table defines how AA-SBC proxies registrations. It maintains a table of multicast destinations, to determine which peer(s) to proxy to when a REGISTER request arrives. The registration routing table handles incoming requests—which peer do we expect to receive a call from, and therefore need to share REGISTER information with?

When AA-SBC receives a REGISTER, it retrieves the directory associated with the request URI to see if it matches any registration-plan from the registration routing table.

If there is no matching in the registration plan, then the REGISTER is checked against the local registration service. If the local registration service determines the REGISTER is accepted, then Eclipse responds with a “200 OK” message to the client. Otherwise, Eclipse responds with a “600 Decline” message to the client.

Registration plan objects

For a detailed explanation of registration handling and various examples of AA-SBC registration-plan and related configuration, see the *Net-Net OS-E – Session Services Configuration Guide*.

Understanding registration service routing tables

AA-SBC has internal routing tables that manage the registration service and call handling data. These tables are:

- the registration routing table, which handles REGISTER requests (described in this chapter)
- the location routing (or call routing) table, which handles INVITE requests (described in Chapter 21, “Dial Plan Objects”).

When a REGISTER comes into the system, the request is forwarded based on the settings reflected in the registration routing table. Use the **show registration-plan** command to view all configured entries; use **show registration-routing** to display active entries.

Registration routing table

The registration routing table defines how AA-SBC proxies registrations. It maintains a table of multicast destinations, to determine which peer(s) to proxy to when a REGISTER request arrives. The registration routing table handles incoming requests—which peer do we expect to receive a call from, and therefore need to share REGISTER information with?

The following is a sample registration routing table. Entries are compiled from registration plans, tag matches, and domain matches (all configured in the server object). Any server listed in the table must be a SIP registrar (server type sip-registrar).

```
NNOS-E> show registration-routing

plan-name     type     match             peer-name    action   hits
---------     ----     ---------------    ---------    ------   ----
abc.com       tag      aster              abc          delegate 0
as.works.net  tag      bw                 works        delegate 0
123.com       tag      123                123          delegate 0
978           phone    sip:1978.*@.*      12xyz        delegate 86
240           phone    sip:240.*@.*       works        delegate 74
```

```
xyz            domain   sip:.*@.*xyz\.com  xyz-server   delegate 36
tele.com       domain   sip:.*@tele\.com   China        delegate 0
work           domain   sip:.*@.*work\.net work         delegate 40
```

The following table describes each field and how its value is derived.

Field name	Description	Derived from...
plan-name	The proxy registration plan for a server.	The plan created with the **registration-plan** object.
type	The portion of the request to match on—user, peer domain, or peer directory tag. If the REGISTER matches the portion identified by the type, the system forwards the request to that server.	A type of *url* or *phone* indicates the entry was made via the registration-plan property. Types of *tag* or *domain* indicate entries from the auto-tag-match and auto-domain-match **server** properties.
match	Regular expression or tag that identifies the “to directory” mapping.	The registration-plan property configures “from” and “to” URL regular expressions.
peer-name	A statically entered peer. This is a configured server of type sip-registrar.	One or more of the following properties, set by the **server routing-type** property: • provider-peer • internal-peer • external-peer

Field name	Description	Derived from...
action	The action that the system is configured to perform for a REGISTER. This is the first step in the handling process. When the system receives a REGISTER, it first attempts to match the request against all delegation peers. If there is any match, the REGISTER is forwarded on. If not, the system walks all non-delegation peers. For each peer, it matches all registration plans according to the configured order. If a match is found for the peer, the system proxies the registration to the peer.	The registration-service peer configuration and server type. If it is a delegation server, then the action is *delegate*. If the server is a SIP registrar, then the action is *proxy*. For any other type, the peer does not appear in the registration routing table.
hits	The number of times this peer/ registration plan has matched a REGISTER request, and the system proxied the request.	This is a counter internal to AA-SBC.

Registration plan object summary

The following table lists and briefly describes the **registration-plan** object. See the following chapter for other objects in the CLI hierarchy:

- Chapter 75, “Virtual System Partition (VSP) Objects”
- Chapter 62, “Session Configuration Objects”
- Chapter 15, “Condition List Objects”

registration-plan

Purpose

Opens the registration plan object through which you define the phone numbers and suffix entries and, optionally, the exceptions for outgoing phone calls.

Syntax

```
config vsp registration-plan
```

Properties

None

settings

Purpose

Creates an exception to a registration match. These settings allow you to override the default lookup behavior in the pattern precedence match. (See Pattern Match Precedence for a complete description.) For example, you can configure domain-aware phones so that a specific range of characters are not subject to the phone-prefix match. Enter a regular expression to create phone numbers that are matched on the suffix found in the URI instead of the phone number. For example, if the URI contained a three-digit extension, and the domain-aware-phone-expression was set to \d\d\d, AA-SBC would match on the domain suffix instead of phone number.

Syntax

```
config vsp registration-plan settings
```

Properties

Property name	Description
domain-aware-phone-expression *regExp*	Overrides the phone number match and causes the system to match on the suffix instead for any calls matching the regular expression. To configure all numbers to be matched on suffix, enter **.*** as the regular expression. **Example: set domain-aware-phone-expression \d\d\d** The default setting is **^$** (null string).
alpha-numeric-phone-expression *regExp*	Allows alphanumeric characters in the SIP URI, and changes the lookup behavior. By default, all phone numbers are subject to a phone prefix match lookup. If the USER field of the SIP URI contains alphanumerics, the system does not treat the URI as a phone number. However, if the characters match this regular expression, then the user *is* treated as a phone number and is subject to a phone prefix match lookup. If it does not match the regular expression, or this property is not set, then the user is not subject to a prefix match lookup. Instead, the system proceeds with a domain match lookup. **Example: set alpha-numeric-phone-expression ^eng[12].*** The default setting is **^$** (null string).

normalization

Purpose

Initiates normalization for matching SIP REGISTER, SUBSCRIBE, and NOTIFY requests. Call normalization is the process of changing all or a portion of a SIP URI so that the SIP call can be matched properly in order to be routed to a particular destination. This normalization occurs before arbitration or routing lookup (facilitating the lookup). Messages in which the REGISTER requests To, From, or Request URI and those that contain the criteria defined in the **match** property are subject to normalization.

Syntax

```
config vsp registration-plan normalization string
```

Properties

Property name	Description
description *string*	Associates a text string with the normalization plan configuration. The string displays in some event logs and status providers to help identify the target. **Example: set description E911server** There is no default setting.
admin {enabled | disabled}	Enables or disables the current normalization plan. If **enabled**, the system applies these settings if there is a registration plan match. **Example: set admin disabled** The default setting is **enabled**.
match *type string*	Specifies what to match in the USER and/or HOST fields in the SIP header in order for the system to apply the normalization plan to calls containing the prefix. Select the type of match to make and then enter a string to match on. **Example: set match domain-exact abc.com** There is no default setting.

Property name	Description
condition-list-match-secondary {false | true}	Specifies whether a condition list match should also be required for the specified match property. If the match property is set to something other than condition-list, you can set this property to **true** to use a condition list in addition to the type selected. In that case, the call must match both the primary key and the condition list. **Example: set condition-list-match-secondary true** The default setting is **false**.
priority *value*	Specifies an order of preference for this normalization plan. Often, a number or URI will match multiple normalization entries. By default, the system uses the most specific match. Use this property to override that default behavior and set a preference based on the **match** property. See Assigning Priority for more information. **Example: set priority 50** Enter any number greater than 1; the default priority setting is **100**.

Property name	Description
user-normalization *settings*	Sets the type of normalization that the system applies to calls to or from a provider (to the USER field of the Request URI). **Example: set user-normalization prepend-to 10 1978** The default type setting is **no**.
apply-to-methods *messageType*	Specifies the message type to which the system applies the **match** property. Those messages containing the configured match in the selected header(s) are then normalized according to this plan. When you modify this value, the system overwrites the current setting with only the message types you specify. For example, if set to the default (all selected) and you enter **NOTIFY**, the system will match against only the NOTIFY portion of the header. Enter multiple message types separated by a plus sign (+) with no spaces. Note that this property is overridden by the values configured with the **sip-message-plan** property of the **settings** object. If a message type is assigned to **dial-plan** in that property, you cannot apply normalization here. **Example: set apply-to-methods REGISTER** The default setting is **REGISTER+SUBSCRIBE+ NOTIFY+PUBLISH**.

arbiter

Purpose

Configures an ordered set of rules to influence the routing arbitration decision for those requests meeting registration plan match criteria. These rules configure different metrics, which AA-SBC uses to select where to forward inbound requests and whether or not to load balance them.

The arbiter function is usually subscriber based. This means that arbitration is applied based on who originated the request (the source), for example, indicated in the URI in the From header. In addition, when a subscriber match is found, the arbiter can allow matching of the request-URI or to-URI (defined in the server configuration, **call-matching-on** property). Note that by using a **condition-list** you can match on either source or destination of a request.

AA-SBC uses the longest prefix/suffix match for lookups within the **arbiter** rules. See Finding the Most-Specific Entry for more information.

If the system does a lookup in the arbitration table and finds no entries, it uses “factory” default settings. These are:

- Use the **best-match** setting for **arbiter-apply**
- Use the most-preferred routing calculation algorithms.

Syntax

```
config vsp registration-plan arbiter string
```

Properties

Property name	Description
description *string*	Associates a text string with the arbiter configuration. The string displays in some event logs and status providers to help identify the target. **Example: set description E911server** There is no default setting.
arbiter-apply {best-match | joined-matches}	Specifies whether to apply the arbiter rules to the best match or to all matches in the routing table lookup. Select either: • **best-match**—by default, the arbiter rules apply to the most specific match. Or, if configured, to the route with the lowest priority (set with the **priority** property of the **route** object). In the event of a tie, the system selects the most specific. • **joined-matches**—the system merges all routes that match the dial plan and then sorts them according to the routing arbitration rule specified in the **rule** property. This value is then used by the **handle-response** property of the **server-pool** object, when set to **try-next-route**. **Example: set arbiter-apply joined-matches** The default setting is **best-match**.
max-call-hunting-options *integer*	Specifies the maximum number of gateways and/or trunk-groups that the system can hunt for a call in case of gateway/trunk failure. **Example: set max-call-hunting-options 50** Enter any number greater than 1; the default setting is **100**.

Property name	Description
call-hunting-type {none | sequential | parallel}	Determines the order or method in which the system forwards the call to the next-hop gateway. Unless set to **none**, this setting takes precedence over any forking settings set by the **server** object **call-hunting-type** property. • **none**—the system forwards the call to the latest binding for the Request URI. • **sequential**— if there are two or more servers in a server pool, the system first tries the primary and then the secondary. • **parallel**—when the system receives a call, it creates two call legs and forwards to both the primary and secondary server. When one server responds, the system disconnects the call with the other server. **Example: set call-hunting-type none** The default setting is **none**.
call-routing-on {request-uri | to-uri | as-is}	Specifies whether the system does routing or location lookups based on the Request URI, the To URI, or an alternate setting. By default, the system performs lookups on the Request URI. Change this setting, for example, when routing information is not available in the Request URI but it is available in the To URI. This setting can also be configured in the **server-pool** object. If values are set in both this and the server-pool, the arbiter settings take precedence. The system does the lookup on either: • **request-URI**—the Request URI, which contains the hop-by-hop destination for the call. • **to-uri**— the To URI, which contains the final destination of the call. • **as-is**—the Request URI (the default) or the value set for this property in the **server-pool** object. **Example: set call-routing-on request-uri** The default setting is **as-is**.

Property name	Description
min-calls-apply-constraints *integer*	Specifies a minimum number of calls that must be active before quality constraints are applied. The system does not route based on quality metrics until the severs and routes have reached the minimum set with this property. **Example: set min-calls-apply-constraints 150** Enter any number between 1 and 65535; the default priority setting is **100**.
max-cost {unlimited | *centsPerSecond*}	Sets the maximum rate, in cents-per-second, that a call can cost. If this property is set to unlimited, there is no limit to the calling rate. **Example: set max-cost 9** Enter a number between 1 and 65535; the default setting is **100**.
min-available-bandwidth *kbps*	Sets a maximum threshold of available bandwidth for a server, limiting the amount of traffic forwarded to that downstream server. For each connected call to a server, the system calculates the bandwidth used based on the CODEC. If all calls to server exceed this bandwidth limit, the system ceases routing to that server until the bandwidth again becomes available. A value of 0 disables the functionality. **Example: set min-available-bandwidth 1000** The default setting is **0** kilobits-per-second.
call-routing-lookup {dial-plan | carrier *carrierReference*}	Specifies which table(s) the system should use for route lookup when routing a call. You can enter multiple routing lookup options. The system searches the route in all tables specified, and then selects a route based on the criteria specified in the arbiter configuration. If you select **dial-plan**, the system performs a call routing table lookup. If you select **carrier**, the system performs a lookup in each referenced carrier table. **Example: set call-routing-lookup carrier “vsp carriers carrier 1"** The default setting is **dial-plan**.

Registration plan objects

Property name	Description
session-config *sessionConfigReference*	Specifies a previously configured entry in the **session-config-pool** object. If this property is set, the system applies the session configuration characteristics to all calls matching this arbiter entry. **Example: set session-config "vsp session-config-pool entry 1"** There is no session configuration applied by default.
apply-to-methods *messageType*	Specifies the message type to which the system applies the **subscriber-match** property. Those messages containing the configured match in the selected header(s) are then subject to routing arbitration according to this plan. When you modify this value, the system overwrites the current setting with only the message types you specify. For example, if set to the default (all selected) and you enter **OPTIONS**, the system will match against only the OPTIONS portion of the header. Enter multiple message types separated by a plus sign (+) with no spaces. Note that this property is overridden by the values configured with the **sip-message-plan** property of the **settings** object. If a message type is assigned to **dial-plan** in that property, you cannot apply routing arbitration here. **Example: set apply-to-methods REGISTER+SUBSCRIBE** The default setting is **INVITE+REGISTER**.

Property name	Description
rule {most-preferred | registration-balance}	Enters rules into the arbiter configuration. If you do not set any rules, the system uses the default settings. Select either: • **most-preferred**—The system uses the server you selected by configuring the server preference. That value is set with the server-pool **server** object **preference** property. If there are multiple carriers marked most-preferred with the same preference, the system uses the next rule in the arbiter to make a forwarding determination. • **registration-balance**—The system uses the server that is selected by the load balancing algorithm. When selected, the system participates in load-balancing of REGISTER requests. Balancing is done in proportion to the maximum number of requests allowed on a server (set by the **server max-number-of-registrations** property). In order for load-balancing across servers to work, you must enable the process globally using the **server-registration-balance** property of the **settings** object. Note that once a REGISTER has been forwarded to a particular server, all future messages intended for that AOR will be forwarded to the correct server. **Example: set rule most-preferred** There is no default setting.

Property name	Description
subscriber-match *type*	Specifies what to match in the USER and/or HOST fields of the FROM URI in order for the system to apply the plan configuration to requests containing the prefix. The **subscriber-match** value defines the criteria for matching entries in the arbitration table; the applicable arbiter is then applied to matches, determining the calculation the system performs. See the Match type options and descriptions table for information on each property option. **Example: set subscriber-match server "vsp enterprise server lcs lcs-server"** The default setting is **phone-prefix** (with no prefix specified).
condition-list-match-secondary {false | true}	Specifies whether a condition list match should also be required for the **subscriber-match** property. If the match property is set to something other than condition-list, you can set this property to **true** to use a condition list in addition to the type selected. In that case, the call must match both the primary key and the condition list. **Example: set condition-list-match-secondary true** The default setting is **false**.
admin {enabled | disabled}	Enables or disables this arbiter entry and its associated session configuration. **Example: set admin disabled** The default setting is **enabled**.
priority *value*	Specifies an order of preference for this registration-plan arbiter entry. Often, a number or URI will match multiple arbiter entries. By default, the system uses the most specific match. Use this property to override that default behavior and set a preference based on the **subscriber-match** property. See Assigning Priority for more information. **Example: set priority 50** Enter any number greater than 1; the default priority setting is **100**.

route

Purpose

Configures AA-SBC to make call routing/forwarding decisions based on information in the To URI. Use the **source-route** object to make routing decisions based on the IP packet header or the From URI of the SIP message. With the source-route object, a route is selected based on the source while this object selects based on the destination. AA-SBC checks for a source-route match first, then a route match.

The **route** configuration specifies the portion of the To URI (dial prefix, domain suffix, condition list criteria) to match on to initiate direction of the call to a particular gateway. If an outgoing call matches the **to-uri-match** value specified in the entry, AA-SBC applies the session configuration entry to the call.

AA-SBC uses a longest-prefix match lookup to match the most specific entry. If a gateway becomes unavailable, the system finds the next longest match and forwards the call to that gateway. See Finding the Most-Specific Entry for a more detailed explanation. Many of the properties in this object can also be set in the **location-service > address-of-record** (or **settings**) objects. The **address-of-record** settings take precedence, however, as they are based on a more specific match. These **route** settings are next, and finally, the **settings** configuration serves as a default in the event of no other match.

Integrating with advanced call-forking based host features

AA-SBC can act as a location NAT device in cases where multiple endpoints share the same AOR (for example, a phone on a desk and one in the lab both registered to jdoe@abc.com). When acting as a location NAT, if the system receives a REGISTER from a particular location, it modifies the request by putting its own IP address in the Contact field and a unique port number in the port field. By doing this, each binding looks unique to the SIP gateway. (Normally, AA-SBC puts the same port number in each request, typically 5060.) This enables you to implement third-party advanced host features that run on top of call forking.

To configure this form of NAT, you must set the following in this object:

- enable the **trunk-port-per-binding** property
- set the port range with the **alter-contact** property (note that port numbers can be shared between AORs)

- set the **action** property to **delegate**

When configured, any REGISTER request that matches this registration **route** or **source-route** plan is sent to the upstream server at one of the included trunk ports. The next REGISTER for the same AOR but from another location will be registered with a different trunk port. (In other words, when a route or source-route plan match is made, AA-SBC creates individual entries for each binding within an AOR.) When AA-SBC receives a call and finds from the location cache entry that **multiple-bindings-nat** is enabled, it will search for the binding with the matching trunk local port and will forward the call.

Configurable actions for registrations

The system can be configured to handle registrations based on matches in the registration plan or against location service **address-of-record** configuration. When AA-SBC receives a registration from an endpoint or an AOR, it can be configured to take on the following available actions, as described in the following table:

Action	Description
accept	The system accepts the registration locally, functioning as a registrar. Note that you must select this option when configuring secure trunking with the system as a proxy.
delegate	The system forwards the REGISTER to an upstream SIP proxy (provider) and resets the contact to itself. This allows an upstream registrar to forward subsequent messages to the system rather than to the phone client directly. When the system receives a 200 OK from the upstream registrar for the delegated REGISTER request, it saves the binding into its location database. Be certain to set the **peer** property if you set the action to delegate.
forward	The system forwards the REGISTER, unchanged, to the server specified in the header.
redirect	The system sends a response to the client with instructions to resend the REGISTER to a different server. The response includes alternative registrars' contact information (the server you configure with the **peer** property of the **route** object).

Action	Description
tunnel	The system creates an OC client-to-LCS server tunnel, via the registration plan, that you can then load balance across. See Configuring Load Balancing Accross OC Client -to-LCS Server Tunnels for complete configuration requirements for load balancing
discard	The system silently discards REGISTER requests matching this registration plan or AOR.
block	The system rejects calls matching this registration plan or AOR with either a “603 Declined” message or with the text of the **response-code** and **response-string** properties, if both are configured. The system sends a “200 OK” response code by default, so be sure to change it to indicate a block if you have configured a response string. WIthout a string, the system sends a “603 Declined.”

Syntax

```
config vsp registration-plan route string
config vsp registration-plan source-route string
```

Properties

Property name	Description
description *string*	Associates a text string with the route configuration. The string displays in some event logs and status providers to help identify the target. **Example: set description E911server** There is no default setting.
admin {enabled | disabled}	Enables or disables the current route or source-route entry. If **enabled**, the system applies these settings if there is a registration plan match. **Example: set admin disabled** The default setting is **enabled**.

Property name	Description
condition-list-match-secondary {false | true}	Specifies whether a condition list match should also be required for the specified match property. If the match property is set to something other than condition-list, you can set this property to **true** to use a condition list in addition to the type selected. In that case, the call must match both the primary key and the condition list. **Example: set condition-list-match-secondary true** The default setting is **false**.
priority *value*	Specifies an order of preference for this registration-plan entry. Often, a number or URI will match multiple entries. By default, the system uses the most-specific match. Use this property to override that default behavior and set a preference based on the **to-uri-match** (route object) or **source-match** (source-route object) properties. **Example: set priority 50** Enter any number greater than 1; the default priority setting is **100**.
action {accept | delegate | forward | redirect | tunnel | discard | block}	Specifies how the system processes any registration it receives that matches the registration-plan route or source-route entry. See Configurable Actions for Registrations for a description of each action. Be certain to set the **peer** property if you set the action to delegate. **Example: set action redirect** The default type setting is **forward**.
peer {none | server reference | exchange reference}	Specifies to which gateway server the system should forward the call. Enter the name of a previously configured server. **Example: set peer server “vsp enterprise servers sip-gateway companyABC”** There is no default setting.

Property name	Description
location-match-preferred {up-to-outbound-peer | best-effort | exclusive | except-from-server | no}	Specifies how the system should forward a call if it finds a location cache match for the endpoint. Select either: • **up-to-outbound-peer**—if the next-hop peer is a **provider** (set with the **server-pool-admission-control > service-type** property), the system forwards the call to that provider peer (provided that the call is not originated from that peer). If it is of **service-type** external or internal, and there is a location cache match for the endpoint, the system forwards the call directly to the endpoint. If there is no match, it forwards the call to the next-hop peer. • **best-effort**—the system always attempts, regardless of the **service-type** setting, to forward the call to the endpoint if there is a location cache match, or to the next-hop peer. If calling to the endpoint fails or times out, the call may, if enabled in the **call-hunting-type** property of the **arbiter**, be sequentially forked to the next-hop peer. • **exclusive**—the system always attempts, regardless of the **service-type** setting, to forward the call to the endpoint if there is a location cache match, or to the next-hop peer. If calling to the endpoint fails or times out, the call is *not* sequentially forked to the next-hop peer, even if sequential forking is enabled. • **except-from-server**—if the call was received from a server, the system tries the location cache first for an endpoint match. Otherwise, it uses the dial-plan to forward the call. • **no**—the system never forwards the call directly to the endpoint (even if it has a location match for it), regardless of the **service-type** setting. **Example: set location-match-preferred best-effort** The default setting is **up-to-outbound-peer**.

Property name	Description
peer-expiration *seconds*	Specifies the value the system writes to the expire time in the REGISTER request, before sending it to the peer. When doing registration delegation, the system changes the expiration value to the specified **peer-expiration** when delegating it to the upstream server. The system saves the expiration value recorded in the 200 OK from the upstream server to its location cache. If you enter 0, the client value remains. Use this property in conjunction with the **default-max-client-expiration** property (or **max-client-expiration**) if **registration-throttling** is set. By setting this, you can offload registration activity from the peer to the system. **Example: set peer-expiration 0** The default setting is **86400** seconds.
authentication-interval *seconds*	Specifies how frequently the system re-authenticates an endpoint. Once an endpoint has registered, the system throttles future registrations, acting as a proxy for the registrar. This property sets how frequently the system lets a REGISTER through to re-authenticate the endpoint. **Example: set authentication-interval 30000** The default setting is **86400** seconds.
default-max-client-expiration {as-requested | *seconds*}	Overwrites the client binding expiration time, as found in the client REGISTER request. The time you enter specifies the maximum time (in seconds) to elapse before a client REGISTER request becomes invalid and the registration information is removed from the location cache. If you enter 0, or as-requested, the client value remains. This value is used unless a more specific match on the request is made with the settings of the **max-client-expiration** property. Use this property in conjunction with the **peer-expiration** property if **registration-throttling** is set. If the client offers a value lower than that configured here, the system uses the lower value. **Example: set max-client-expiration 5400** The default setting is **3600** seconds.

Property name	**Description**
max-client-expiration *protocol* {any | NAT | no-NAT} *seconds*	Sets an expiration timer for client bindings based on the transport protocol and whether or not the source of the REGISTER request is behind a NAT device. The system only applies the new timer to the session if both the protocol and the NAT firewall conditions are met. If these conditions are not met, the system bases the client binding timeout value on the setting of the **default-max-expiration** property. This timer setting applies to the protocol: • **any**—regardless of the presence or lack of a NAT device. • **NAT**—only if a NAT device is present. • **no-NAT**—only if a NAT device is not present. **Example: set max-expiration TLS NAT 1800** The default setting is **3600** seconds.
min-client-expiration {as-requested | *seconds*}	Overwrites the client's minimum expiration time, as found in the client REGISTER request. The time you enter specifies the minimum time (in seconds) to elapse before a client REGISTER request can become invalid and the registration information can be removed from the location cache. If you enter 0, or as-requested, the client value remains. **Example: set min-client-expiration 30** The default setting is **15** seconds.

Property name	Description
alter-contact {no | local-host [*localPort*] [*maxTrunkPorts*] | local-maddr [*localPort*] [*maxTrunkPorts*] | trunk-port-per-aor | trunk-port-per-binding | trunk-port-per-endpoint}	Specifies whether to change certain portions of the request CONTACT field in the incoming URI. If you do choose to modify the URI, you can optionally specify a local port. The port number you enter appears as the port from which the REGISTER was sent, and becomes the contact port number for the AOR binding. By setting this, subsequent calls will use this local port. In addition, if you are enabling **trunk-port-per-binding**, you set the maximum number of ports the system can allocate to a single AOR. The local port assignment sets the beginning of the range, and the maximum trunk ports sets how many port numbers are included in the range. This value must be less than or equal to the value set with the **max-bindings-per-aor** property. Note that port numbers can be shared between AORs. See Integrating with Advanced Call-Forking Based Host Features for a full description. *continued*

Property name	Description
alter-contact *continued*	Select one of the following: • **no**—no modification takes place. • **local-host**—alters the Host portion of the incoming URI to the system host information (local interface address). • **local-maddr**—appends the MAddr parameter with the address set to the system local address, while setting the Contact URI to be the same as the To header. • **trunk-port-per-aor**—Provides a unique IP address and port for each registered endpoint. When set, the endpoint will show a unique transport address to the upstream server/ gateway, instead of the interface IP address and well-known port of the system. Use this, for example, for admission control on the upstream server/gateway. If using this setting, you must enable the **mirror-all-entries** property of the master services **registration** object. Note that assigned trunk ports are only valid for UDP transport. • **trunk-port-per-binding**—Sets the system to allocate a trunk port number or reuse a previous trunk port number in the Contact header for the matching binding. The system allocates the port from the pool of available ports set with the **media-ports** object. Note that assigned trunk ports are only valid for UDP transport. • **trunk-port-per-endpoint**—Sets the system to allocate a single unique trunk port for all devices that are behind the same device (devices that present the same remote host IP address, contact IP address, and port). This would be the case, for example, with an ATA that hosts multiple lines. The system allocates the port from the pool of available ports set with the **media-ports** object. Note that assigned trunk ports are only valid for UDP transport. **Example: set alter-contact no** The default setting is **local-host**. The default local port setting (for local-host and local-maddr) is 5060.

Property name	**Description**
apply-to-methods *messageType*	Specifies the message type to which the system applies the **to-uri-match** (route object) or **source-match** (source-route object) property. Those messages containing the configured match in the selected header(s) are then directed according to this plan. When you modify this value, the system overwrites the current setting with only the message types you specify. For example, if set to the default (all selected) and you enter **NOTIFY**, the system will match against only the NOTIFY portion of the header. Enter multiple message types separated by a plus sign (+) with no spaces. Note that this property is overridden by the values configured with the **sip-message-plan** property of the **settings** object. If a message type is assigned to dial-plan in that property, you cannot control message direction here. **Example: set apply-to-methods NOTIFY** The default setting is **REGISTER+SUBSCRIBE+NOTIFY+ PUBLISH**.
session-config *sessionConfigReference*	Specifies a previously configured entry in the **session-config-pool** object. If this property is set, the system applies the session configuration characteristics to all calls matching this registration plan entry. **Example: set session-config vsp session-config-pool entry 1** There is no session configuration applied by default.

Property name	Description
registration-throttling {enabled | disabled}	Sets whether the system responds locally to a registration request. When **enabled**, if the system has an existing binding for the contact in the REGISTER request and that binding has not expired on the peer, the system forgoes registration. When throttled, the system also does not perform proxy authorization. If **disabled**, the system forwards all registration requests to the delegate server. **Example: set registration-throttling disabled** The default setting is **enabled**.
binding-replacement {loose | tight | strict | custom}	Determines whether or not a subsequent REGISTER for a binding can replace the previous entry by setting match criteria for the new REGISTER request. If any of the parameters do not match, the system creates a new binding. After the system has delegated a REGISTER request, it will only process the response if it finds a valid binding for this response. When the following conditions are met for an option setting, the system replaces a binding if: • **loose**—the previous binding matches the public IP address. • **tight**— the previous binding matches the public IP address, port number, and transport protocol. • **strict**—the previous binding matches the public IP address, port number, and transport protocol, and the private IP address and port number. • **custom**—the previous binding matches the user-specified elements. **Example: set binding-replacement custom public-ip+public-port+call-ID** The default setting is **strict**.

Property name	**Description**
max-bindings-per-AOR {*value* | none}	Specifies the maximum number of bindings allowed for each AOR. If set to **none**, this value is derived from the **max-bindings-per-AOR** property in the location service **settings** object. If you set a value, that value takes precedence over the location service **settings** for matching routes. However, the **address-of-record** object **max-bindings** value has the highest precedence, if a call should match the specific AOR and a registration-pl an. Note that the maximum number of trunk ports, set with the **alter-contact** property, must be less than or equal to this value. **Example: set max-bindings-per-AOR 3** The default setting is **none**.

Property name	Description
validate-bindings {loose | tight | strict}	Specifies how the system validates bindings received from the delegate server and whether it strips unknown or unrequested contacts from the 200 OK going back to the UAC. When set, the system does one of the following: • **loose**—sends all contact headers in the REGISTER response on to the UAC without validation. • **tight**—validates the contacts listed in the REGISTER request against the location cache. The system then strips any contact that is not listed in the location cache or was not requested by the UAC before sending it on. If **registration-throttling** is enabled (any setting other than **no**), the system sends all registered bindings for that AOR in throttled REGISTER responses. • **strict**—stores the contact headers sent to the delegate and uses them to validate against the contacts received in the REGISTER response. This prevents problems where a registrar returns all the bindings associated with an AOR instead of only the bindings associated with a particular call ID. If a REGISTER response contains contacts that were not in the original REGISTER request sent to the delegate by the system, or if there are any duplicate bindings, the system discards them. When a binding is discarded, it is not added to the location cache and it is stripped from the REGISTER response prior to being sent to the UAC. **Example: set validate-bindings strict** The default setting is **loose**.
route object only: to-uri-match *type string*	Specifies what to match in the USER and/or HOST fields in the SIP header in order for the system to apply the entry normalization plan to calls containing the prefix. Select the type of match to make and then enter a string to match on. **Example: set to-uri-match domain-exact abc.com** The default type setting is **phone-prefix**. There is no default for the prefix itself.

Registration plan objects

Property name	Description
source-route object only: source-match *type*	Specifies the match criteria for the source of the SIP message. The system sets the next-hop server (defined with the **peer** property) for all traffic that matches this configured source. **Example: set source-match ipnet 192.168.0.0/16** There is no default setting.
edp [NAT] [TCP] [TLS]	*Secondary property.* Sets the connection type that the Expiration Discovery Process (EDP) is being used with, either NAT, TCP, and/or TLS. EDP is the process the system uses to detect a maximum of time in which system can reach an endpoint as indicated by the location binding, regardless of the expiration time set by the endpoint. With NAT, the selected expiration time keeps the NAT pinhole continually open for the endpoint—a firewall otherwise may age out a pinhole more quickly than the binding expiration. With TCP or TLS, the selected expiration time keeps the connection refreshed regularly and continually open for the endpoint. Otherwise, a TLS connection may age out because of TCP socket inactive timeout. **Example: set edp NAT** There is no default setting.

Property name	Description
edp-expire-grow *seconds*	*Secondary property.* Specifies the number of seconds that the edp-expiration timer sent in the 200 OK message should increase or decrease by when the EDP process cycle receives a response to the system's OPTIONS message from an endpoint. When the EDP process is triggered, the edp-expiration timer starts, and the system changes a binding's state to WAITING. When the timer expires, the system sends an OPTIONS message to the endpoint and changes the state to PINGING. When the endpoint responds, the system changes the state to PINGED and the edp-expiration timer value is incremented by the value of **edp-expire-grow**. If the endpoint does not respond, the state is changed to TIMEOUT and the value of edp-expiration is decreased by the **edp-ping-timeout** value. (The value used in the previous cycle was the correct expiration time for the binding.) **Example: set edp-expire-grow 15** The default setting is **10** seconds.
edp-ping-timeout *seconds*	*Secondary property.* Specifies the number of seconds added to the **min-client-expiration** value to set the EDP expiration time that is sent in the 200 OK message. If that new value (the sum of **min-client-expiration** and **edp-ping-timeout**) is less than the original expiration, the system triggers the EDP process. **Example: set edp-ping-timeout 45** The default setting is **30** seconds.
response-code *code*	*Secondary property.* Sets the response code that the system sends to an endpoint when the **action** property is set to **accept** or **block**. (2xx response codes indicate success; change this value if the action is **block** and you have configured a **response-string**.) **Example: set response-code 201** The default response code is **200**.

Property name	Description
response-string *string*	*Secondary property.* Sets the response string that the system sends to an endpoint when the **action** property is set to **accept** or **block**. **Example: set response-string “REGISTER was blocked”** There is no default response string.
session-linger *seconds*	*Secondary property.* Specifies the number of seconds a registration session remains active while awaiting reregistration in response to a challenge. By keeping the session active, subsequent REGISTERs from an endpoint that are responding to a “401 Auth Challenge” can reuse the same session, improving registration authentication performance. **Example: set session-linger 10** Enter a value from 0 to 60; the default setting is **0** seconds.
calling-group *callingGroupReference*	*Secondary property.* References a calling-group configuration that is applied to REGISTERs matching this registration plan route entry. The **calling-groups** configuration applies admission control to matching calls and creates a way to segregate routing arbitration, call routing, policy, and normalization based on the user group. **Example: set calling-group “vsp calling-groups group 1”** There is no calling group association by default.

Property name	Description
client-type {windows-messenger | office-communicator	*Secondary property.* Sets the type for the client end of a client-to-LCS Server tunnel. This setting should be the same as the *fromServerReference* setting, for the appropriate transport, in the **sip** object. This setting is only used for load balancing across OC client-to-LCS server tunnels. The CLI displays other options for client type, but you must select either windows-messenger of office-communicator. See Configuring Load Balancing Across OC Client-to-LCS Server Tunnels for detailed configuration information. **Example: set client-type windows-messenger** The default setting is **unknown**.

Property name	Description
access-control-level {strict | tight | loose}	*Secondary property.* Specifies the information a user must match in an existing binding to be considered "known" by the system. Any call coming in to the system (and handled by this registration plan) that does not meet the criteria is considered an unregistered sender. That call is then handled according to the setting in the **unregistered-sender-directive** property of either the **pre-session-config**, **server**, or **carrier** object (if the call came from that server or carrier). This property applies to message types other than REGISTER requests. On successful registration, the system saves this requirement level setting, as part of the binding, in the location cache. When receiving a future call, the system performs a location cache lookup on the From URI to determine if it matches the necessary characteristics for the indicated requirement level. The registration is valid: • **strict**—when the previous binding matches the IP address, transport protocol, port number, and socket of the new binding. • **tight**— when the previous binding has the same IP address as the new binding. • **loose**—when the same AOR was registered previously. In most cases, the system declines any call that fails the corresponding level of registration requirement check. However, if you have set the **unregistered-sender-directive** to **allow**, the system will pass the call even if it fails the requirement check. **Example: set access-control-level loose** The default setting is **tight**.

Property name	Description
allow-proxy-bindings {enabled | disabled}	*Secondary property.* Specifies whether to allow an endpoint to register on behalf of a second IP address. For example, an endpoint might register with a second contact listed in the REGISTER request. When **enabled**, the system accepts the registration and writes both bindings to the registration database. When **disabled**, the endpoint can only register with its own IP address. **Example: set allow-proxy-bindings enabled** The default setting is **disabled**.
uac-preferred-contact {auto | public | private}	*Secondary property.* Determines where the Host portion of the INVITE Request URI or To header is derived from. Select either: • **auto**—the Host portion is determined automatically. If a REGISTER is received from a SIP proxy, then the host is set to private.. • **public**—if the caller is behind a firewall, the Host portion is set to the public IP address of the firewall (the NAT address). • **private**—if the caller is behind a firewall, the Host portion is set to the private IP address of the UAC. **Example: set uac-preferred-contact private** The default setting is **auto**.

source-route

Purpose

Configures AA-SBC to make call routing/forwarding decisions based on information in the IP packet header or the From URI of the SIP message. (Use the **peer** object to make routing decisions based on To URI information.) With the route object, a route is selected based on the destination while this object selects based on the source. AA-SBC checks for a source-route match first, then a route match.

The source-route configuration specifies the portion of the IP header or From URI to match on to initiate direction of the call to a particular gateway (set with the **peer** property). If an outgoing call matches the **source-match** value specified in the entry, AA-SBC applies the entry session configuration to the call.

Syntax

```
config vsp registration-plan source-route string
```

Properties

See the **route** object for property descriptions.

proxy

Purpose

Sets the criteria to determine for which calls AA-SBC should act as proxy. As a proxy, AA-SBC provides SIP registration, location, policy, and other services that determine the outcome of the SIP call. AA-SBC terminates the registration, responds with a 200 OK, and saves the binding to the location database. The system then generates a new REGISTER to the upstream registrar. Use the **peer** object to identify servers to which the system proxies registrations.

Proxy operations

To determine whether to proxy a registration, AA-SBC evaluates the contents of a client AOR. (Contrast this with the **route** or **source-route** entries, which route calls based on the contents of the REGISTER requests.) The system only proxies calls for previously registered users (AORs). A user is registered if any of the user aliases have previously registered. So, after initial successful registration, the system creates a location cache entry for the alias—a binding in the AOR.

When a REGISTER request arrives at AA-SBC, the following happens:

1. After successful registration, AA-SBC creates a location cache entry for the alias.This triggers an event.
2. For this event (the contact by the alias), AA-SBC does a lookup in the location cache to determine whether the alias was proxied previously. If it was previously registered, AA-SBC does not proxy a new REGISTER, as the database between between proxy and peer is updated daily. (Set with the **server** > **request-download** property)
3. If the alias was not proxied previously, AA-SBC does a location cache lookup to determine whether the user is registered. If the user is registered, the system does a proxy plan lookup.
4. If there is a match on a proxy plan, AA-SBC proxies the registration (on behalf of the alias) to all configured peers for that match.

Syntax

```
config vsp registration-plan proxy name
config vsp registration-plan source-proxy name
```

Properties

Property name	Description
description *string*	Associates a text string with the proxy configuration. The string displays in some event logs and status providers to help identify the target. **Example: set description E911server** There is no default setting.
admin {enabled | disabled}	Enables or disables the current proxy or source-proxy plan. When **enabled**, the system applies these settings if the AOR matches the criteria set with the **uri-match** (proxy) or **source-match** (source-proxy) properties. **Example: set admin disabled** The default setting is **enabled**.
proxy object only: uri-match *type*	Sets match criteria for the AOR. For all REGISTERs in which a lookup results in a matching AOR, the system applies this proxy plan. **Example: set uri-match domain-suffix abc.com** There is no default setting.
source-proxy object only: source-match *type*	Sets match criteria for the AOR. For all REGISTERs in which a lookup on the source field results in a matching AOR, the system applies this source proxy plan. **Example: set source-match server “vsp enterprise servers lcs ABC-server”** There is no default setting.
priority *value*	Specifies an order of preference for this proxy plan. Often, a number or URI will match multiple entries. By default, the system uses the most-specific match. Use this property to override that default behavior and set a preference based on the **uri-match** property. The lower the priority value, the higher the preference. **Example: set priority 50** Enter any number greater than 1; the default priority setting is **100**.

peer

Purpose

Identifies the upstream server(s) to which AA-SBC proxies registrations. You can configure any number of peers per proxy configuration; when a REGISTER matches the **proxy** or **source-proxy** match criteria, it is forward to all configured peers for that plan. See Proxy Operations to understand when AA-SBC proxies a REGISTER request. Use this configuration to mirror registrations to multiple servers. You can also create a configuration to delegate to a primary server and then mirror using the proxy if authentication is not in use.

Enter an integer as an index identifier for the peer.

Syntax

```
config vsp registration-plan proxy name peer ID
config vsp registration-plan source-proxy name peer ID
```

Properties

Property name	Description
age-notification-only {true | false}	Specifies whether the system sends an UNREGISTER request to the configured peer when a binding expires or is removed. This field is only applicable when **vsp > registration-proxy** is enabled. **Example: set age-notification-only true** The default setting is **false**.
broadcast-ownership {true | false}	Enables AA-SBC to broadcast a subsequent UNREGISTER request for a contact to a group of non-clustered AA-SBC peers. This is applicable when the REGISTER was delegated and the system had a binding installed for the contact. **Example: set broadcast-ownership true** The default setting is **false**.
alter-uri {none | next-hop-ip | next-hop-domain | host [*hostName*] | directory *directoryReference*}	Specifies how to change the To/From field of the URI (for outbound traffic) so that it is acceptable to the next-hop server. See Altering URIs for more information. **Example: set alter-uri next-hop-domain** The default setting is **none**.

Property name	Description
contact no contact {local-host | local-maddr} [*localPort*] [*maxTrunkPorts*] [no | rinstance | *string*] contact {trunk-port-per-aor | trunk-port-per-binding | trunk-port-per-endpoint}	Specifies whether to change certain portions of the REQUEST Contact field in the incoming URI. If you do choose to modify the URI, you can optionally specify a local (trunk) port. The port number you enter appears as the port from which the REGISTER was sent, and becomes the contact port number for the AOR binding. By setting this, subsequent calls for this binding will be received at this local (trunk) port. Use this property to differentiate multiple line appearances. Leave the optional maxTrunkPorts setting at 1, the default. Select one of the following: • **no**—no modification takes place. • **local-host**—alters the Host portion of the incoming URI to the system host information (local interface address) or the information you specify. Optionally, you can set the system to R-instance (a unique ID) to the Contact URI. • **local-maddr**—appends the MAddr parameter, with the address set to the system local address, while setting the Contact URI to be the same as the To header. • **trunk-port-per-aor**—provides a unique IP address and port for each registered endpoint. When set, the endpoint will show a unique transport address to the upstream server/switch, instead of the interface IP address and well-known port of the system. Use this, for example, for admission control on the upstream server/switch. If using this setting, you must enable the **mirror-all-entries** property of the **master-services > registration** object. **Example: set contact no** The default setting is **local-host**. The default local port setting (for local-host and local-maddr) is 5060.

Property name	Description
peer {none | server *trunkReference* | carrier *carrierReference* | gateway *gatewayReference* | hunt-group *huntGrpReference*}	Identifies the peer(s) to which the system proxies registrations for each proxy or source-proxy plan. Enter an upstream destination type and reference to a configured entity of that type. For the **server** option, enter a server of type sip-gateway or sip-connection. **Example: set peer gateway “vsp carriers carrier ABC gateway xyz”** The default type is **server**.

source-proxy

Purpose

Creates a proxy plan for REGISTER requests based on the source field of the URI. See Proxy Operations for a complete description of how AA-SBC handles proxy registration. Use the **peer** object to identify servers to which the system proxies registrations.

Syntax

```
config vsp registration-plan source-proxy name
```

Properties

See the **proxy** object for property descriptions.

Registration plan objects

56. Registration service objects

Registration service object description

The **registration-service** object configures the local AA-SBC registration service that provides address-of-record updates to the location service database, making AA-SBC the registrar. The location database can be imported and exported over SIP REGISTER sessions with other registration peers. This object is secondary.

See Chapter 38, “Location service objects” for information on configuring address or records and the location service database.

Registration service object summary

The following table lists and briefly describes the **registration-service** objects. See the following chapter for other objects in the CLI hierarchy:

- Chapter 75, “Virtual system partition (VSP) objects”

Object name	Description
`registration-service`	*Secondary object.* Enables and sets expiration times for a specified registration service.

registration-service

Purpose

Sets the registration service operational settings. This registration service is a registrar that can process REGISTER requests and add them to the location services database. It can also forward REGISTER requests and database entries to peer registrars. AA-SBC declines REGISTERs if **registration-service** is not configured and the registration object in the session configuration is administratively disabled.

Syntax

```
config vsp registration-service
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables this registration service. If disabled, the system rejects any REGISTER request sent to the registration service. **Example: set admin enabled** The default setting is **enabled**.
max-expiration {as-requested | *seconds*}	Overwrites the client maximum expiration time, as found in the client REGISTER request. The time you enter specifies the maximum time (in seconds) to elapse before a client REGISTER request becomes invalid and the registration information is removed from the location cache. If you enter 0, or **as-requested**, the client value remains. **Example: set min-expiration 3600** The default setting is **as-requested**.

Property name	Description
min-expiration seconds {as-requested | *seconds*}	Overwrites the client minimum expiration time, as found in the client REGISTER request. The time you enter specifies the minimum time (in seconds) to elapse before a client REGISTER request becomes invalid and the registration information is removed from the location database. If you enter 0, or **as-requested**, the client value remains. **Example: set min-expiration 100** The default setting is **as-requested**.
session-linger seconds {as-requested | *seconds*}	*Secondary property.* Specifies the number of seconds a registration session remains active while awaiting reregistration in response to a challenge. By keeping the session active, subsequent REGISTERs from an endpoint that are responding to a “401 Auth Challenge” can reuse the same session, improving registration authentication performance. **Example: set min-expiration 100** Enter a value from 0 to 60; the default setting is **5**.

Registration service objects

57. Routing objects

Routing description

The route object allows you to manually create static IP routes to destination networks and hosts (routers) connected to the Internet. A static route provide a constant route to a specific network or host router. This static route takes precedence over dynamically learned routes and is not overwritten by dynamic routing protocols (such as RIP and OSPF) running in your network.

AA-SBC uses a static route when its routing table does not have a route to other devices in the network. By defining a default route, AA-SBC can send traffic to other devices in the network even if you do not define any other routes. You configure static routes for each IP interface that would benefit from the functionality.

Routing object summary

The following table lists and briefly describes the **routing** objects. See the following chapters for other objects in the CLI hierarchy:

- Chapter 14, “Cluster, box, and interface objects”
- Chapter 35, “IP objects”
- Chapter 77, “VLAN objects”

Object name	Description
`routing`	Opens the routing configuration object for editing.
`route`	Configures a named static route to a destination IP network or host, or creates a default route (0.0.0.0) in the network routing table.

routing

Purpose

Opens the routing configuration object for editing. The **routing** object allows you access to the object that creates one or more static routes that are added to the system routing table.

Syntax

On a public IP interface:

```
config cluster box integer interface ethX ip name routing
config cluster vrrp vinterface vxID ip name routing
config cluster box integer interface ethX vlan integer ip name routing
config box interface ethX ip name routing
config box interface ethX vlan integer ip name routing
```

Properties

None

route

Purpose

Adds to or edits a static route in the system routing table. These static routes provide a constant route to a destination host (router) or destination network that connects to the public Internet. Or, when creating a static route within a virtual firewall, you are creating entries that configure AA-SBC to be able to reach subnets within the private network. For any static route you configure, you also define the local router or gateway as the next-hop router to the destination. You can also configure a gateway for a default route; this is the route AA-SBC uses when no other entries match the destination.

AA-SBC uses the route configuration to determine how to resolve destination addresses. The match is always on the most specific address available. Use the **show routing** command to display AA-SBC routing table.

Enter a name for the route. This name appears in the routing table display. You can create as many static routes as you wish, but each must be created in an individual route object.

Syntax

On a public IP interface:

```
config cluster box integer interface ethX ip name routing route name
config cluster box integer interface ethX vlan integer ip name routing
   route name
config cluster vrrp vinterface vxID ip name routing route name
config box interface ethX ip name routing route name
config box interface ethX vlan integer ip name routing route name
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables this route configuration. **Example: set admin disabled** The default setting is **enabled**.
destination (network *ipAddress*/*mask* | host *ipAddress* | default}	Identifies the destination address that this object represents. You must also configure a corresponding **gateway** for the route. The system resolves the destination to the gateway you specify. Destination network and host addresses are added to the system routing table. If a destination does not match any other route in the routing table, the system uses the **default** route (0.0.0.0/0) and its corresponding gateway. The default route directs any data addressed to any network numbers that are not explicitly contained in the routing table. Create a static route entering one of the following: • **network**— an IP address and mask to match the destination network. • **host**— a host IP address • **default**—creates 0.0.0.0/0. **Example:** **`set destination default`** **`set destination network 192.168.124.0/24`** **`set destination host 192.168.124`** The default setting is the **default** route. However, without a gateway configured, this value is not functional.

Property name	Description
gateway *ipAddress*	Sets the gateway or next hop IP address for the packet. **Example: set gateway 192.168.124.6** The default setting is 0.0.0.0.
metric *preference*	Associates a cost with the static route that the system adds to its services route and route DB tables. The lower the metric the more preferred the route. The system chooses the more preferred route when there are multiple interfaces available on the same network. **Example: set metric 10** Enter a value between 0 and 4294967295; the default setting is **1**.

58. Secure Shell objects

SSH description

Secure Shell (SSH) Server Version 2 on AA-SBC provides secure client/server communications, remote logins, and file transfers using encryption and public-key authentication. To establish a secure connection and communications session, SSH uses a key pair that you generate or receive from a valid certificate authority (CA). AA-SBC uses the OpenSSH daemon for SSH support.

An SSH session allows you to transfer files with Secure Shell File Transfer Protocol (SFTP), providing more secure transfers than FTP and an easy-to-use interface. SSH uses counters that record SFTP activity over the SSH connection.

When running SSH on AA-SBC, the SSH session is transparent and the CLI appears just as it would if you were connecting from a console or over Telnet. The AA-SBC implementation of SSH does not support all the user-configurable attributes typically supported by SSH workstations. If you try to change an attribute that AA-SBC does not support, you will receive a notification that the setting failed.

SSH object summary

The following table lists and briefly describes the SSH object. See the following chapters for other objects in the CLI hierarchy:

- Chapter 14, “Cluster, box, and interface objects”
- Chapter 77, “VLAN objects”
- Chapter 35, “IP objects”

Object name	Description
`ssh`	Configures AA-SBC SSH communications settings on the current interface.

ssh

Purpose

Configures an SSH listener on an IP interface. Note that although you can configure SSH settings on each IP interface, there is only one SSH daemon running, with one configuration. The SSH configuration is an aggregate of the separate interface SSH configurations.

Syntax

```
config cluster box number interface ethX ip name ssh
config cluster box number interface ethX vlan number ip name ssh
config box interface ethX ip name ssh
config box interface ethX vlan number ip name ssh
```

Properties

Property name	Description
admin {enabled | disabled}	Sets the administrative state of the SSH protocol, either **enabled** (running) or **disabled**. When disabled, the parameters of SSH can still be configured, but do not become active until **admin** is set to **enabled**. **Example: set admin disabled** The default setting is **enabled**.
max-sessions *integer*	Sets the maximum number of concurrent SSH sessions allowed, enforced at the box level. The enforced value is an aggregate of the SSH session limits set on each IP interface that has SSH enabled. For example, to enforce a limit of five total SSH sessions per box, you could set IP “A” to an SSH session limit of two and IP “B” to an SSH session limit of three, for a total of five. **Example: set max-sessions 4** Enter a value between 1 and 32. The default number of concurrent sessions is **8**.
idle-timeout *integer*	Specifies the amount of time (in seconds) allowed to elapse before the system closes the SSH session due to inactivity. **Example: set idle-timeout 300** Enter a value between 60 and 86,400. The default setting is **600** seconds (10 minutes).

Property name	Description
port *integer*	Identifies the known TCP port through which the system listens for SSH sessions. **Example: set port 25** Enter a value between 1 and 65535. The default SSH port is port **22**.
mode {ssh-1 | ssh-2 | compatibility}	Sets the version of SSH the system should use. Be aware multiple vulnerabilities exist in SSH version 1, and it is therefore not secure. Use the **compatibility** setting to allow the system to determine the version in use by the remote system and set its own version to match. **Example: set mode ssh-2** The default setting is **compatibility**.
authentication {password | public-key | rsa}	Sets the authentication method(s) AA-SBC uses to authenticate users, either **password** or public key. To use public key SSH authentication on AA-SBC, generate a public/private key pair, install the public key on the system, and install the private key on your SSH client. You can select either SSH version 1 (**RSA**) or SSH version 2 (**public-key**) authentication. You can select If you do not specify any authentication methods, the system applies the OpenSSH defaults. **Example: set authentication password** There is no default setting.

Property name	Description
account {root | ssh}	Sets the account(s) to use for SSH authentication, either root or a user account (ssh). Use the **ssh password** action to set up an account password if the account type is set to **ssh**. By default, the system uses the root account. However, if you set the account type to **ssh**, the root account no longer applies. You can then add it back in using this property. **Example: set account root** There is no default setting, but the system uses root if not otherwise set.
log-level {quiet | fatal | error | info | verbose | debug | debug1 | debug2 | debug3}	Specifies to the SSH daemon the level of SSH events to generate and send to the AA-SBC event log. The SSH component sends all events of that level and higher. **Example: set log-level debug** The default setting is **verbose**.

59. Server objects

Server description

Enterprise services are SIP-enabled real-time communication systems and collaboration services. By configuring AA-SBC to recognize a particular enterprise service, you are drawing that service under the security protection of AA-SBC, preventing application-level attacks. These services allow an organization to support, among others:

- IP PBX hosted VoIP services
- enterprise instant messaging systems
- mobile devices
- presence-based applications

Enterprise services work by establishing an application (directory) server, a SIP component in the enterprise. Client programs access the server to look up user entries, and the server expects a certain set of users to be using it. For example, a server might be an IBM/Lotus Sametime server. Using AA-SBC, you would configure a link between that server and a directory containing the Sametime users.

Specifically, AA-SBC supports the following enterprise servers:

- IBM Lotus Sametime Server
- Microsoft Live Communications Server (LCS) 2005
- Nortel Multimedia Communications Server (MCS)
- Avaya IP telephony PBX
- A generic SIP source/destination
- A generic SIP registration server
- A DNS group
- A SIP connection
- An H.323 gateway

Note: While you can configure directory services at any time, you must enable the master-services directory object for AA-SBC to use the service. See Chapter 39, “Master services objects” for more information.

For detailed information on AA-SBC enterprise gateways, refer to the *Net-Net OS-E – Session Services Configuration Guide*.

Normalization in the servers group

The server pool server-pool-admission-control objects provides inbound and outbound normalization settings to apply to calls going to or from the server. Use outbound-normalization for calls destined for a server; use inbound-normalization for calls received from the server. The objects properties are common for servers, gateway, and trunk groups, and are described in Chapter 46, “Outbound and inbound normalization objects”.

Server descriptions

The following sections briefly describe each server type that is supported by AA-SBC.

Sametime description

The IBM Lotus Instant Messaging and Web Conferencing (Sametime) is the IBM real-time collaboration platform. It is used to manage the flow of instant messages, streaming audio and video, shared applications, and whiteboard sessions. Sametime provides instant access to people and information through integrated presence awareness and brings together centralized and geographically dispersed participants.

LCS description

Microsoft Office Live Communications Server (LCS) is an instant messaging (IM) and real-time collaboration solution that enables enterprises to reach, collaborate, and respond to information more quickly than e-mail and telephone services. LCS uses SIP, SIP Instant Messaging and Presence Leveraging Extensions (SIMPLE), and the Real-Time Transport Protocol. In an environment where LCS is in use, there is also an Active Directory in use. Most of the LCS configuration is looking at LCS-specific attributes in the Active Directory. Generally, you'd configure either LCS or Active Directory, but not both.

MCS description

Nortel Multimedia Communication Server (MCS) is a SIP-based communications and collaborative applications platform. Using industry-standard hardware, the MCS integrates voice with video, collaboration, and presence services. With MCS, service providers can offer video and voice calling, call screening and routing and call management, multipoint media conferencing, Web application collaboration, instant messaging, white boarding, file exchange, and presence.

Avaya description

The Avaya Converged Communication Server (CCS) integrates SIP telephony with existing voice networks to provide a range of services to subscribing users and devices. Using a standards-based architecture, CCS integrates SIP registrar, proxy, presence, instant message gateway, and instant messaging to provide full communications in a multivendor environment.

SIP gateway description

Configures a generic SIP server. For example, it could be a SIP proxy, a SIP application server, or a PSTN gateway. By configuring the public switched telephone network (PSTN) gateway, you can configure AA-SBC to allow enterprises to continue call operations even if their provider server is busy or down. The way that AA-SBC handles unavailable servers and future call routing is controlled by the local-mode setting of the **routing-settings** property. See the routing-setting attribute descriptions for more information.

Specify the SIP URI for the gateway, in the form SIP:*gatewayIdentity*. For example, SIP:sip-server@broadsoft.com.

SIP host description

The SIP host is a generic server description that allows AA-SBC configuration to include a server configuration for a non-explicit server type.

DNS group description

Dns-group is a server configuration template for servers that do not use a server pool configuration because they can be resolved by DNS. When AA-SBC receives a REGISTER request, if the domain is the same as that configured for a dns-group, AA-SBC clones the configuration of that dns-group for the server. AA-SBC then does three DNS lookups—NAPTR, SRV, and A—to resolve the transport protocol, port, and address. (If multiple records are found, AA-SBC uses the preference set in the DNS server to select the primary.) AA-SBC then adds the server to the server pool. If the domain from the REGISTER is different from the dns-group, AA-SBC creates a new server object and clones the configuration from the dns-group. Note that you must configure a dial plan and/or registration plan to point to the **dns-group**.

SIP connection description

The SIP connection server type provides a client/server model between AA-SBC and customer premise equipment. AA-SBC fills the server role, while the connection (line) between the CPE and AA-SBC acts as client. This connection may be a single line, a shared line, or a group of shared lines to the enterprise or a residence. The point of connection on a shared line (the CPE) represents one or multiple direct inward dial (DID) numbers. Behind the CPE, however, may be many more endpoints. In this configuration, the client initiates, or re-establishes in the event of failure, the connection with AA-SBC.

Using this server type allows you to create a configuration specific to an AOR. For instance, it allows you to control the number of concurrent calls to (emission control) and from (admission control) the specific AOR. You can override the global location cache settings that set the number of concurrent calls, and allow more or fewer calls based on the connection.

Additionally, AA-SBC can learn client transport information through dynamic registration. Within the registration-plan, you can reference a **sip-connection** type server. Then, when a REGISTER comes in from the CPE (sip-connection server) and matches a registration-plan, when AA-SBC installs a location cache entry, it saves the sip-connection name and reference in the location entry. If the sip-connection has unknown transport information (host, port, transport, local port and so on), AA-SBC can use the dynamic learn feature (if enabled), to derive the sip-connection transport information from the client registration.

H.323 gateway description

H.323 is a widely-deployed multimedia conferencing protocol which includes voice, video, and data conferencing for use over packet switched networks. AA-SBC acts as a peer Gatekeeper on a H.323 system, supporting Gatekeeper-Routed Signaling or direct endpoint signaling. This object configures an H.323 gateway; use the h323 object to enable H.323 on an interface and set the listening ports. AA-SBC supports H.323-to-SIP, SIP-to-H.323, and H.323-to-H.323 calls.

CLI hierarchy information

See the following chapters for other objects in the CLI hierarchy:

- Chapter 75, “Virtual system partition (VSP) objects”
- Chapter 27, “Enterprise objects”

Server object summary

The following table lists and briefly describes the **servers** objects.

Object name	Description
`server`	Configures the parameters for communication between the enterprise directory server and AA-SBC.
`h323-to-sip-fromheader-spec`	Specifies how to build a SIP From header from an H.323 SETUP message.
h323-to-sip-toheader-spec	Specifies how to build a SIP To header from an H.323 SETUP message.

Object name	Description
h323-reason-translate	Maps H.323 reason codes to SIP response values.
h225-settings	
h245-settings	
q931-cause-sip-response-map	
q931-settings	
federator	*Secondary object.* Creates a federation identity.
default-sip-settings	*Secondary object.* Configures SIP communications settings for calls destined for the specified server.
server-pool	Adds servers to the server pool and defines connection characteristics.
server-pool-admission-control	Adds servers to the server pool.
outbound-normalization	Applies normalization settings to outbound calls sent to this server. See Chapter 46, “Outbound and inbound normalization objects”, for a full description.
inbound-normalization	Applies normalization settings to inbound calls received from this server. See Chapter 46, “Outbound and inbound normalization objects”, for a full description.
registration-proxy	Sets characteristics of a relationship between registration peers.
network	Sets server socket values.
ccs	Configures AA-SBC to recognize the Avaya CCS.

server

Purpose

Opens the server configuration object to allow setting the parameters for communication between the directory server and AA-SBC, supporting the following enterprise services:

- IBM Lotus Sametime Server (sametime)
- Microsoft Live Communications Server 2005 (lcs)
- Nortel Network Multimedia Communications Server (mcs)
- Avaya IP telephony PBX (avaya)
- A generic SIP source/destination (sip-host)
- A PSTN gateway (sip-gateway)
- DNS group
- SIP connection
- H.323 gateway

AA-SBC uses strict, tight, or loose matching rules to map. A REGISTER request or INVITE must match according to what you have configured within this object. By default, AA-SBC uses strict rules for mapping, meaning that it only maps to names that contain an exact match of the domain name you entered. If you have configured the **domain-alias** property, AA-SBC uses tight rules, meaning it will map on either the name or alias. If you set the **domain-subnet** property, loose matching rules are in effect.

Note: While you can configure directory services at any time, you must enable the master-services directory object for AA-SBC to use the service. See Chapter 39, “Master services objects” for more information.
Note: When creating or editing a SIP gateway, specify the SIP URI for the gateway, in the form SIP:*gatewayIdentity*. For example, SIP:sip-server@broadsoft.com.

Routing-setting definitions

The **routing-setting** property allows you to select one or more server attributes. The following table describes each of these attributes in detail.

Attribute	Description
normalization	When AA-SBC receives a request (e.g., an INVITE or REGISTER), it checks the host portion of the request. When **normalization** is enabled, if the host portion matches the domain name, domain name alias, the subnet, or a server-pool entry, AA-SBC changes the host name to the server domain name. By making this change, AA-SBC can then match the request on a configured dial or registration plan.
auto-tag-match	When enabled, if the server has a configured directory, AA-SBC automatically creates a dial plan and registration plan for the server.
auto-domain-match	When enabled, AA-SBC creates a domain-based dial and registration plan for the server. The plan uses the domain-exact **request-uri-match** type (matches any USER field and a HOST field containing the exact domain name specified). AA-SBC uses the domain name configured for the server, and the resulting plans have no normalization or session configuration. The action associated with the plans is **delegate**.
pstn-backup	When a server is down (not reachable), if **pstn-backup** is not selected, AA-SBC changes the state of the server to “not available.” Any dial or registration plan with reference to that server is removed from the call routing or registration routing table. In its normal state, AA-SBC operates in provider mode, forwarding calls to a provider application server. If the server fails, and AA-SBC has location information for the provider, it forwards calls locally. Otherwise, AA-SBC forwards calls to a PSTN gateway. You configure the gateway using the **pstn-gateway** server object. This is called local mode. When enabled, an unavailable server state changes to “local mode.” Plan entries stay in the routing tables.

Attribute	Description
outbound-association	When enabled, AA-SBC uses its management system to derive associations when originating a SIP message. When disabled, AA-SBC sends the message straight through, which results in better performance.
cxc-from	When enabled, AA-SBC changes the From header to the AA-SBC local identity when proxying registrations to an upstream server. When disabled the original URI remains in place, meaning that the REGISTER is derived directly from the sender.
local-mode	Sets AA-SBC to always function in local mode. In *provider mode*, the normal state, AA-SBC forwards calls to a provider application server. If the server has failed, and AA-SBC has location information for the provider, it forwards calls locally. Otherwise, AA-SBC forwards calls to a PSTN gateway. You configure the gateway using the sip-gateway server object. This is called *local mode*. AA-SBC detects provider failure using the **failover-detection** property setting of the server-pool-admission-control object. When local mode is not selected, the system stays in local mode until the it determines that the server has resumed functionality. When the server again becomes available, AA-SBC reverts the registrar peer back to provider mode and retries calling through the provider. If the call is successful, AA-SBC stays in provider mode.

Service-type definitions

The **service-type** property allows you to set the way AA-SBC handles INVITE and REGISTER requests and database exchanges. The following table describes each of these settings in detail.

Attribute	Description
provider	Specifies the server as a provider peer, which means that AA-SBC proxies INVITE and REGISTER requests. If a peer has proxy-registration configured, then AA-SBC proxies the registration. In other words, AA-SBC intercepts the REGISTER, stores the contact information in the location cache, and generates a new request with AA-SBC as the contact. If the peer does not have proxy-registration configured, AA-SBC does not proxy the registration. Instead, it checks the call routing table to see if the request URI matches a provider. If there is a match, AA-SBC forwards the request to the peer listed in the table. If there is not a match, AA-SBC walks the call routing table, entry by entry. If a match is found in the table, the INVITE is forwarded to the peer. Otherwise, the INVITE is forwarded to the default outbound proxy.

Attribute	Description
internal	Specifies the server as an internal peer (internal to the enterprise or subscriber community under a single service provider). If the internal peer is a SIP registrar, then AA-SBC does location database exchanges with it. When an INVITE matches a dial plan for an internal peer, AA-SBC first looks up the location cache for forwarding information. If found, the INVITE is forwarded to the location binding. Otherwise, the INVITE is forwarded to the internal peer.
external	Specifies the server as an external peer (external to the enterprise or subscriber community under a single service provider). If the external peer is a SIP registrar, then AA-SBC challenges the REGISTER request with RADIUS or DIAMETER, and if successful, then passes the request on to the external peer. AA-SBC stores the location binding in the cache in case the external peer forwards future INVITEs. When an INVITE matches a dial plan for an external peer, AA-SBC first looks up the location cache for forwarding information. If found, the INVITE is forwarded to the location binding. Otherwise, the INVITE is forwarded to the external peer.

Syntax

```
config vsp enterprise servers sametime string
config vsp enterprise servers lcs string
config vsp enterprise servers mcs string
config vsp enterprise servers avaya string
config vsp enterprise servers sip-gateway SIP:gatewayIdentity
config vsp enterprise servers h323-server string
config vsp enterprise servers sip-host string
config vsp enterprise servers dns-group string
config vsp enterprise servers sip-connection string
```

Properties

Property name	Description
description *string*	Associates a text string with a server configuration. The string displays in some event logs and status providers to help identify the target. **Example: set description E911server** There is no default setting.
admin {enabled | disabled}	Specifies whether the system uses this server in the current session. If **enabled**, the system uses this server. If **disabled**, the system does not use this server. **Example: set admin enabled** The default setting is **enabled**.
carrier *string*	Associates a text string with a server. The string can later be used to group and categorize servers. **Example: set carrier server1** There is no default setting.
domain *domainName*	Identifies a domain to be used by the system for server normalization. In cases where the server is associated with: • a single domain—enter that domain. • multiple domains—enter one of the domain names. • no domain—enter another valid domain on the system. (This might be the case with a PSTN gateway for example.) Also, you must set this property if you enable the settings **local-directory-based-user-services** property without configuring the **directory** property (to assign a directory to a server). Set this domain name to match user SIP addresses to the appropriate server (by use of the domain). **Example: set domain voip.companyABC.com** There is no default setting.

Property name	Description
routing-tag *string*	Controls which outbound interface SIP traffic uses. The routing-tag indicates the interface on the server where a SIP message with a matching routing-tag would be forwarded. The SIP message derives its routing-tag from the session config or IP interface classification-tag, depending on the configuration scenario. This property sets the initial routing tag for a server. If there is a policy match that applies to the server, and that configuration sets a routing tag (with the routing-settings ingress- and egress-classification-tag), the policy setting takes precedence. **Example: set routing-tag lcs1** There is no default setting.
failover-detection {none | auto | ping | register}	Determines the method to use to detect a when a upstream server peer is unavailable (and has resumed availability). Select either: • **none**—the system does no checking, and the server peer always appears available, even when down. • **auto**—the system uses an internal algorithm to count transaction failures. If a message to the server fails, the system resends the message the number of times defined in the sip-settings object **max-retransmissions** property. When the system reaches the retransmission threshold, it increments the **dead-threshold** count and then starts the retransmission process again. When the server reaches the failure threshold (set with the **dead-threshold** property), the system changes the server state to DOWN and sends no further requests. The fallback timer (set with the **dead-fallback-interval** property) activates. *continued*

Property name	Description
`failover-detection` *continued*	When the timer expires, the system decrements the server dead count by one and can again send requests to the server. If it receives no response, the system again increments the count and reaches the threshold, restarting the process. If the server responds, the system decrements the dead count again, until the count reaches 0. Note that if there is a major transport error, such as “no socket,” the system skips the retransmission step and increments the **dead-threshold** count. Use this setting in the case where a server does not respond to SIP OPTIONS messages. You must also enable the vsp **auto-server-fail-detection** property (which acts like a master switch) when using this option. • **ping**—the system uses the sip-ping utility to check server availability. SIP ping sends SIP OPTIONS messages to a peer. When enabled, the system pings its peers at an interval defined in the **ping-interval** property. If the peer is not operational (determined by the dead-threshold property, the system switches to local mode if pstn-backup is checked (**routing-setting** property), or to unavailable mode if pstn-backup is not checked. When the system is again able to successfully ping the peer, it reverts to provider mode. • **register**—the system determines server availability by sending a REGISTER request to the server. If there is no response from the provider, the server is assumed down. Note that you must configure a user for the server, with the appropriate password. (See the **user** and **password-tag** properties). **Example: set failover-detection auto** The default setting is **none**.

Property name	Description
failover-termination {enabled | disabled}	Sets whether calls are disconnected if AA-SBC detects server failure. If the **failover-detection** property is enabled (set to anything other than **none**), and a server fails, AA-SBC terminates all calls going through that server when this property is **enabled**. When **disabled**, connections are unaffected. **Example: set failover-termination enabled** The default setting is **disabled**.
domain-alias *domainName*	Sets the system to recognize an alias domain as the domain in which the server resides. You can enter as many aliases as you choose. **Example: set domain-alias eng.companyABC.com** There is no default setting.
domain-subnet *ipAddress/mask*	Sets the IP subnets serviced by this server. **Example: set domain-subnet 1.2.3.4/16** There is no default setting.
local *ipAddress*	Sets the server local IP address. **Example: set local 192.168.1.4** There is no default setting.
ping-interval *seconds*	Sets the number of seconds between ping packets sent between the system and the SIP registrar server. **Example: set ping-interval 30** The default setting is **10** seconds.
dead-threshold *retransmissions*	Specifies the number of transaction failures (and resulting retransmissions) a server can experience before the server state is changed to DOWN. This threshold is used in the **auto** and **ping** options of the **failover-detection** property. **Example: set dead-threshold 15** Enter a value from 1 to 255; the default setting is **4** retransmissions.

Property name	Description
dead-fallback-interval *seconds*	Sets the fallback timer for the server. During this period, the system does not send REGISTER or INVITES to the down server. After the timer expires, the system decrements the **dead-threshold** by 1. This timer is used in the **auto** and **ping** options of the **failover-detection** property. **Example: set dead-fallback-interval 450** Enter a value from 30 to 65535; the default setting is **300** seconds.
handle-3xx-locally-routing-lookup {enabled | disabled}	Specifies whether the system should do a dial plan lookup on the REQUEST URI of a newly generated INVITE based on a 302 response received from this server. This property works in conjunction with the **handle-3xx-locally** property of the sip-settings session config object. If that property is enabled, the system generates a new INVITE when it receives a 3xx response. The system puts the contents of the CONTACT field in the REQUEST URI of the new INVITE. You should **disable** this feature if your server is configured to explicitly forward the message to a specified third sever. Set this to **enabled** if the message is coming from an endpoint with instructions to forward the message to a different AOR. **Example: set handle-3xx-locally-routing-lookup disabled** The default setting is **enabled**.

Property name	Description
unregistered-sender-directive {allow | discard | refuse [*resultCode*] [*resultString*]}	Sets the action the system takes when it receives a packet with an unknown sender in the "From" field of the INVITE packet. Use the **registration-requirement-level** setting in the route or source-route object to define what is considered unknown. Select one of the following actions: • **allow**—the system permits the packet to proceed toward its destination. • **discard**—the system immediately discards the packet. • **refuse**—the system discards the packet but sends a response to indicate having done so. The response includes an error code (default of 400 but you can enter any value between 400 and 699) and an optional description. **Example: set unregistered-sender-directive refuse 404 "unknown sender"** The default setting is **allow**. If you select **refuse**, the default result code is **400**.
inbound-session-config-pool-entry *sessionConfigReference*	Specifies a session configuration entry to apply to all inbound traffic destined for this server. **Example: set inbound-session-config-pool-entry "vsp session-config-pool entry inboundPolicy"** There is no default setting.
outbound-session-config-pool-entry *sessionConfigReference*	Specifies a session configuration entry to apply to all outbound traffic from or through this server. **Example: set outbound-session-config-pool-entry "vsp session-config-pool entry outboundPolicy"** There is no default setting.
server-type *type*	Sets the server version or function. The type that you select is dependent on the server type that you are configuring. See below for the options for each server:

Property name	Description
sametime	Identifies the server as version 3.1, operating as either a direct or proxy server (SIP connector). Enter one of the following: • sametime-31 • sametime-31-sip-connector • sametime-75 • sametime-75-sip-connector
lcs	Identifies the LCS server version (2003 or 2005) and the function (server-only or access proxy). Enter one of the following: • lcs-2003 • lcs-2003-access-proxy • lcs-2005 • lcs-2005-access-proxy • ocs-2007 • ocs-2007-edge-server
mcs	Identifies the server as Nortel Networks MCS: • nortel-mcs
avaya	Identifies the server as an Avaya PBX: • avaya
sip-host	When using the generic SIP server, identifies the function the server is fulfilling. Enter one of the following: • windows-messenger • sip-proxy • sipura • snom • polycom • office-communicator • nortel-mcp
sip-gateway	Identifies the server as a SIP gateway: • sip-proxy • sipx
dns-group	Identifies a DNS group: • dns-group

Server objects

Property name	Description
sip-connection	Identifies the connection type: • windows-messenger • sip-proxy • sipura • snom • polycom • office-communicator • nortel-mcp
h323-server	Identifies the server as an H.323 gateway: • h323-gw • h323-gatekeeper
peer-identity *sipURI*	*Secondary property.* Specifies a unique URI to identify a remote peer. AA-SBC uses the peer identity (usually found in the FROM header) in peer-to-peer SIP messaging to identify where a SIP message is from. The system can use this information to identify a peer with which to swap location database records. **Example: set peer-identity sip:nnos-e@companyABC.com** There is no default setting.
directory `directoryReference`	*Secondary property.* Creates the link between the server and the name directory it uses. Enter the full path name to a configured directory. Also, if you enabled the settings **local-directory-based-user-services** property**,** you must either set this property (to assign a directory to a server) or set the **domain** property to match user SIP addresses to the appropriate server (by use of the domain). **Example: set directory vsp\enterprise\directories\notes-directory ABCco** There is no default setting.

Property name	Description
user *string*	*Secondary property.* Assigns a user name that the system must supply when challenged by the server (the name of the person qualified to log into this directory server). Enter the name expected by the server, do not create it here. The user name and password-tag (below) are used for authentication between the system and server. **Example: set user admin** There is no default setting. This name must match the username configured on the server.
password-tag *string*	*Secondary property.* Specifies the tag associated with the shared secret used to authenticate transactions between the system and this server. This is the tag associated with the password that the system must supply when challenged by the server. See Using passwords and tags for information on the AA-SBC two-part password mechanism. **Example: set password-tag secure** There is no default setting. This password associated with this tag must match the password configured on the server.
routing-setting *attributes*	*Secondary property.* Sets attributes of the server. See Routing-setting definitions for a description of each option. **Example: set routing-setting auto-tag-match+auto-domain-match** The default setting is **normalization** and **outbound-association**.

Property name	Description
loop-detection {strict | tight | loose}	*Secondary property.* Sets the aggressiveness with which the system enforces call routing loop detection. (The most aggressive requires the fewest parameters to match for the system to drop the call.) Select one of the following: • **strict**—if the system receives a call from a SIP proxy, and a DNS or dial-plan lookup resolves that the source and destination address are the same, the system drops the call. This is the most aggressive. • **tight**—if the system finds the source and destination address, transport protocol, and port to be the same, it drops the call. • **loose**—the system uses standard SIP loop detection (based on the VIA header). When the system finds its own address in the list of SIP proxies traversed, it allows the packet through. **Example: set match strict** The default setting is **tight**.
service-type {provider | internal | external}	*Secondary property.* Specifies the way in which the system handles INVITE and REGISTER requests and database exchanges. See Service-type definitions for complete descriptions of each option. **Example: set service-type internal** The default setting is **provider**.
peer-max-interval *seconds*	*Secondary property.* Specifies the value the system writes to the max-interval setting for a peer. When doing registration delegation, the system changes the expiration value in the REGISTER request to the specified **max-interval** when delegating it to the upstream server. The system saves the expiration value recorded in the 200OK from the upstream server to its location cache. If you enter 0, the peer value remains. **Example: set peer-max-interval 0** The default setting is **86400** seconds (24 hours).

Property name	Description
peer-min-interval *seconds*	*Secondary property.* Specifies the value the system writes to the min-interval setting for a peer. When doing registration delegation, the system changes the expiration value in the REGISTER request to the specified **min-interval** when delegating it to the upstream server. The system saves the expiration value recorded in the 200OK from the upstream server to its location cache. If you enter 0, the peer value remains. **Example: set peer-min-interval 0** The default setting is **3600** seconds.
registration-request-timeout *seconds*	*Secondary property.* Specifies the number of seconds the system waits for a response after sending a REGISTER request to this server. If the system does not receive a response within the configured time, it sends, to the endpoint, notification that service is not available. **Example: set registration-request-timeout 8** The default setting is **10** seconds.
default-policy *`policyReference`*	*Secondary property.* Sets the name of the policy to apply if no more specific policy is in place. Enter a previously configured policy reference. **Example: set default-policy vsp\policies\session-policies\policy lcs** There is no default setting.

Property name	Description
user-group-policy *groupName policyReference*	*Secondary property.* Specifies the policy to apply to users of this server who are members of the specified group. The group can be either a user group from the directory service schema or a virtual group constructed in the configuration for policy application purposes. Enter a group name, and the system applies the specified policy to any user belonging to that group. Also enter the complete path to a previously configured policy reference. If this value is set both here and at the enterprise level, through the enterprise object, the system applies both settings. **Example: set user-group-policy lcsAdmin “vsp policies session-policies policy noIM”** There is no default setting.
to-policy *policyReference*	*Secondary property.* References a policy to apply. If the SIP messages that start a session are directed to this server, the system applies and evaluates the referenced **to-policy**. **Example: set to-policy vsp\policies\session-policies\policy toPolicy** There is no default setting.
from-policy *policyReference*	*Secondary property.* References a policy to apply. If the SIP messages that start a session come from this server, the system applies and evaluates the referenced **from-policy**. **Example: set from-policy vsp\policies\session-policies\policy fromPolicy** There is no default setting.
fork-delay *seconds*	*Secondary property.* Sets the period that the system waits before “ringing” another SIP device registered with a user. **Example: set fork-delay 3** The default setting is **0** seconds (all destinations ring simultaneously).

Property name	Description
server-pool-flush {enabled | disabled}	*Secondary property.* Activates the ability to flush the server pool and relearn the entries from its configuration and the DNS server. The frequency with which the system flushes the server pool is determined by the TTL on the DNS response. **Example: set server-pool-flush enabled** The default setting is **disabled**.
message-filtering {none | st-AD *localDomain federatedDomain*}	*Secondary property.* Specifies a message filter to apply to traffic passing through this server. Do not change this setting from the default, **none**, unless specified to do so by Technical Support personnel. **Example: set message-filtering none** The default setting is **none**.

DNS-group and/or sip-connection properties

Property name	Description
domain-port *portNumber* *dns-group only*	Provides local-port functionality for a server of type dns-group. See the server-pool-admission-control **local-port** property description for details. Enter a value between 1 and 65535; there is no default domain-port number.
host *dns-group and sip-connection only*	See the server-pool-admission-control **host** property description for details.
transport *dns-group and sip-connection only*	See the server-pool-admission-control **transport** property description for details.
port *dns-group and sip-connection only*	See the server-pool-admission-control **port** property description for details.
local-port *dns-group and sip-connection only*	See the server-pool-admission-control **local-port** property description for details.
connection-role *sip-connection only*	See the server-pool-admission-control **connection-role** property description for details.

Property name	Description
admission-control *dns-group and sip-connection only*	See the server-pool-admission-control **admission-control** property description for details.
emission-control *dns-group and sip-connection only*	See the server-pool-admission-control **emission-control** property description for details.
max-bandwidth *dns-group and sip-connection only*	See the server-pool-admission-control **max-bandwidth** property description for details.
max-number-of-concurrent-calls *dns-group and sip-connection only*	See the server-pool-admission-control **max-number-of-concurrent-calls** property description for details.
max-calls-in-setup *dns-group and sip-connection only*	See the server-pool-admission-control **max-calls-in-setup** property description for details.

H.323-specific properties

Property name	Description
fast-start {enabled | disabled}	Specifies whether the system offers H.323 faststart mode in the SETUP message. When enabled, the H.323 network can connect a call with as few as two messages, speeding setup time. Additionally, if all media can be negotiated within those messages, opening an H.245 channel becomes unnecessary. **Example: set fast-start disabled** The default setting is **enabled**.
h245-tunnel {enabled | disabled}	When enabled, data from the H.245 protocol is sent within the H.323 data, preventing the need to open an extra TCP connection for the H.245 data. When **disabled**, they are sent separately. Some applications, such as Microsoft NetMeeting, do not support tunneling. **Example: set h245-tunnel disabled** The default setting is **enabled**.

Property name	Description
early-h245 {reject | notunnel}	Specifies how AA-SBC should handle calls that arrive supporting early H.245. If set to **reject**, the system rejects (and therefore drops) the call. If set to **notunnel**, the system operates as if the **h245-tunnel** property is disabled (an extra TCP connection is opened. **Example: set early-h245 reject** The default setting is **notunnel**.
manual-ringback {enabled | disabled}	Specifies whether the stack or the endpoint application controls sending the alerting messages. When **enabled**, the stack sends the messages on behalf of the endpoint application. When **disabled**, the endpoint application sends them. **Example: set manual-ringback disabled** The default setting is **enabled**.
use-inbound-call-settings {enabled | disabled}	Specifies whether an outbound call uses the h323 server settings or the settings of the inbound call. This property is applicable to H.323-to-H.323 calls only. When **enabled**, the outbound call uses the relevant settings (e.g., fast start and tunneling) of the inbound call. Leave this property set to **disabled** for SIP-to-H.323 calls. **Example: set use-inbound-call-settings enabled** The default setting is **disabled**.
use-h450-hold-retrieve {enabled | disabled}	Specifies how AA-SBC handles hold and retrieve operations for SIP-to-H.323 calls. When this property is enabled, the system sends these operations as H.450 supplemental service messages. When disabled, the system uses H.323 signalling to pause the remote transmitter. **Example: set use-h450-hold-retrieve disabled** The default setting is **disabled**.

Property name	Description
fwd-progress-as-alerting {enabled | disabled}	Specifies the format in which AA-SBC sends progress messages to the destination phone during H.323-to-H.323 calls. When **enabled**, the system sends a progress message (which contains ringtone information) as an alerting message. Use this setting if the terminating gateway does not support progress messages carrying end-to-end (remote) ringback. When **disabled**, the system forwards progress messages in their received format. **Example: set fwd-progress-as-alerting enabled** The default setting is **disabled**.
sip-h323-dtmf-translate {INBAND | RFC2833 | INFO} {INBAND | RFC2833 | Q931 | H245ALPHA | H245SIGNAL}	Specifies the SIP and H.323 DTMF types to advertise to the respective sides. On the SIP side, the system translates DTMF into the type set; on the H.323 side the system advertises the setting as its capability. When setting this property, the methods must match on both sides; both inband (INBAND-to-INBAND or RFC 2833-to-RFC 2833) or both out-of-band (INFO-to-Q.931 or H.245). **Example: set sip-h323-dtmf-translate INFO Q931** The default setting is **INBAND** for both the SIP and H.323 DTMF types.
default-terminal-type *integer*	Specifies the H.245 (multimedia control protocol) terminal type, which is then used in determining the master or slave role of the endpoint. Typically, a value of less than 50 indicates slave operation; a value greater than 200 indicates master operation.. **Example: set default-terminal-type 100** The default setting is **60**.

Property name	Description
multiple-calls {enabled | disabled}	Configures the system to run multiple calls over a single TCP connection. This should be enabled for trunking, allowing all calls from a gateway to the system to be sent over one connection. Use the maintain-connection property to keep the connection up for a period of time after the last call ends. **Example: set multiple-calls enabled** The default setting is **disabled**.
maintain-connection {enabled | disabled}	Specifies whether to leave a connection up after a call (or the last call if trunking is enabled with the multiple-calls property) ends. If this property is enabled, the connection is left in tact until the conn-idle-timeout timer expires. **Example: set maintain-connection enabled** The default setting is **disabled**.
conn-idle-timeout *seconds*	Specifies the number of seconds to wait before tearing down a connection. This setting is only applicable if the maintain-connection property is enabled. **Example: set conn-idle-timeout 2400** The default setting is **3600**.
h323-user-alias {none | dialedDigits | h323ID *ID* | urlID | emailID}	Specifies which type of remote alias the system sets to be sent in the outbound SETUP message. The system uses either: • none—no alias is sent. • dialedDigits—the called-party-number from the SIP side. • h323ID—the specified string, identifying a gateway. • urlID—the To header from the SIP side. • emailID—the To header from the SIP side. **Example: set h323-user-alias emailID** The default setting is **none**.

Property name	Description
q931-bearer-capability-ie [*coding-standard*] [*info-transfer-capability*] [*transfer-mode*] [*tranfer-rate*] [*user-info-laye1-protocol*]	Sets the Q931 Bbearer Capability in SETUP message (for an outbound h323 call) and/or a CONNECT message (for an inbound h323 call). This is the system transfer capability, inserted into a SIP-to-H.323 call, indicating to the network the services it is requesting. The defaults should be changed only to match a specific PBX/switch. If the network does not support the service, the call is rejected. See ITU-T Recommendation Q.931,Annex B for more information. Set: • **coding standard—**CCITTStd, reservedInternationalStd, nationalStd, or locationSpecificStd. • **information transfer capability**—speeck, unrestrictedDigital, restrictedDigital, audio31kHZ, unrestrictedDigitalWithTones, or video. • **transfer-mode—**circuitMode or packetMode. • **transfer-rate—**packetMode, 64Kbps, 128Kbps, 384Kbps, 1536Kbps or 1920Kbps. • **layer-1 protocol—**CCITTStdRate, g711uLaw, g711aLaw, g7221Adpcm, g722g725, h261, nonCCITTStdRate, CCITTStdRatev120 or x31. **Example: set q931-bearer-capability-ie cciTTStd speeck circuitMode 64Kbps x31** **The default settings are CCITTStd, audio31kHz, circuitMode**, **64Kbps**, and **g711uLawis**..
numbering-plan {unknown | ISDN | data | telex | national-standard | private | reserved}	Specifies the format for the numbering plan for called and calling numbers, represented in the first message out. The numbering plan helps to interpret the origin and destination of a call. See ITU-T Recommendation Q.931, Annex B for more information. **Example: set numbering-plan ISDN** The default setting is **ISDN**.

Property name	Description
admission-control {enabled | disabled}	Specifies whether the system applies limitations when forwarding H.323-to-SIP or SIP-to-H.323 calls. The system tracks the number of concurrent incoming calls, and if this property is enabled, the system does not forward H.323 calls between the system and the remote H.323 gateway if the limit has been reached. Instead, it rejects the call via H.323 signaling. If disabled, the system does forward all calls. (Set the call limit with the max-concurrent-h323-calls property.) **Example: set admission-control enabled** The default setting is **disabled**.
max-concurrent-h323-calls *integer*	Specifies the maximum number of concurrent calls between the system and the remote H.323 gateway. When this number is reached, the system applies admission control (if enabled), causing call rejection via H.323 signaling until the value drops below the threshold. **Example: set max-concurrent-h323-calls 2000** The default setting is **1500**.
call-alerting-timeout *seconds*	Specifies the maximum time the H.323 process will wait for a remote H.323 terminal or gateway to respond to a SETUP message. This setting helps AA-SBC reclaim call resources if the remote gateway is not responding to the SETUP message. **Example: set call-alerting-timeout 6** The default setting is **4** seconds.
call-establishment-timeout *seconds*	Specifies the maximum time allowed for a call to become connected. When the timer expires, the system cleanly tears down the H.323 remote gateway connection. **Example: set call-establishment-timeout 90** The default setting is **60** seconds.

Property name	Description
h245-establish-timeout *seconds*	Specifies the maximum number of seconds to wait for the remote gateway to complete the H.245 TCPconnection. If the timer expires, AA-SBC clears the call. **Example: set h245-establish-timeout 2** The default setting is **1** second.
end-session-timeout *seconds*	Specifies the maximum number of seconds to wait after sending an H245 EndSession or H225 ReleaseComplete before clearing the call, if the call did not end gracefully on its own. **Example: set end-session-timeout 90** The default setting is **15** seconds.
use-as-default-gw {true | false}	Specifies whether this server configuration should be used for calls in which an IP lookup does not result in a server assignment. When the system receives a call from an external gateway, it does a lookup on the IP address. If the lookup fails, the system then searches the H.323 server configurations for one with this **use-as-default-gw** property set to **true**. That server configuration is then applied to the call. **Example: set use-as-default-gw true** The default setting is **false**.
presentation-indicator {allowed | restricted | numberNotAvailable | reserved}	Sets the presentation indicator bit in the Q931 privacy header of an outbound H323 setup message. (This bit controls whether a calling number can be displayed to the called party or not.) **Example: set presentation-indicator restricted** Select either allowed, restricted, numberNotAvailable, or reserved. The default setting is **allowed**.
fwd-retrieve-no-tx {true | false}	Specifies whether AA-SBC forwards a Retrieve event if no transmit channel can be opened. If true, the system forwards the event; if false, it does not. **Example: set fwd-retrieve-no-tx false** The default setting is **true**.

Property name	Description
screening-indicator {not-screened | verifiedPassed | verifiedFailed | networkProvided}	Sets the default value AA-SBC sends to the remote H323 gateway in the SETUP message to indicate if the call originator has been “screened” or verified. Select either: • **notScreened—**The caller has not been verified. • **verifiedPassed—**AA-SBC has verified the caller. • **verifiedFailed—**AA-SBC has failed the caller for verification. • **networkProvided—**A network element upstream of AA-SBC has verified the caller. The screening indicator is often used with the presentation-indicator to determine if the caller phone can see the caller-ID. **Example: set screening-indicator verifiedPassed** The default setting is **notScreened**.
privacy-dynamic {true | false}	Specifies whether AA-SBC should translate SIP and H323 values for screening and privacy (instead of using default values). If set to **true**, AA-SBC maps the values for SIP-to-H323 and H323-to-SIP calls. To do so, it uses the Remote-Party-ID header on the SIP side (the screen and privacy tags) and the Q931 CallingPartyIE on the H323 side. **Example: set privacy-dynamic false** The default setting is **true**.
use-progress-inband {enabled | disabled}	Sets whether the calling party uses local or inband ring tone. When inband, the ringing is generated from the called party as audio. Otherwise, the system uses a control signal to generate the ring locally. When this property is **enabled** (the default), the system forwards an indication that ring tone is inband. When **disabled**, the system does not forward the indication so the calling phone generates a local ring tone. **Example: set use-progress-inband disabled** The default setting is **enabled**.

Property name	Description
wait-for-remote-tcs {true | false}	Sets the point at which AA-SBC enters H323 Phase B (exchange of capabilities). When set to **true** (the default), the system waits for the other side to send a “Terminal Capability Set” message. When **false**, the system sends capabilities as soon as it enters Phase B. **Example: set wait-for-remote-tcs false** The default setting is **true**.
forward-retrieve-no-tx {true | false}	Specifies whether AA-SBC forwards a Retrieve event if no transmit channel can be opened. If **true**, the system forwards the event; if **false**, it does not. **Example: set forward-retrieve-no-tx false** The default setting is **true**.
codec-selection {local | remote | followMSD}	*Secondary property*. Sets how AA-SBC chooses a CODEC for a call. The system selects either: • **local**—the highest preference SIP CODEC that H.323 supports. • **remote**—the highest preference H.323 CODEC that SIP supports. • **followMSD**—the CODEC based on the outcome of the master/slave determination (a process that is part of the H.323 handshake). If AA-SBC is the master, the selection is equivalent to the **local** setting; as slave it is equivalent to the **remote** setting. **Example: set codec-selection local** The default setting is **followMSD**.
map-ptime-to-fpp {true | false}	*Secondary property.* Sets the ptime (SIP-side SDP parameter indicating a suggested maximum frames per packet rate) to the H245 FPP parameter (actual frames per packet rate supported by the connection). When set to **true**, AA-SBC sets the suggested rate to the H245 advertised rate. **Example: set map-ptime-to-fpp true** The default setting is **false**.

Property name	Description
map-fpp-to-ptime {true | false}	*Secondary property*. Sets the H245 FPP parameter (actual frames per packet rate supported by the connection) to the ptime (SIP-side SDP parameter indicating a suggested maximum frames per packet rate). When set to **true**, AA-SBC sets the H245 advertised rate to the SDP suggested rate. **Example: set map-fpp-to-ptime true** The default setting is **false**.
reinvite-type {emptyTermCapSet | extendedFastConnect}	*Secondary property*. Specifies how a SIP reinvitation to the H.323 gateway should occur. Set this to match your external H.323 gateway requirements when SIP needs to rebuild the session, for example because something in the SDP changed. **Example: set reinvite-type extendedFastConnect** The default setting is **emptyTermCapSet**.
use-incoming-display-ie {true | false}	*Secondary property*. Specifies whether AA-SBC maps the received DisplayIE field in the Q931 SETUP message to the displayname field in the SIP INVITE From header for outgoing INVITEs. If set to **false**, AA-SBC does not perform the mapping. **Example: set use-incoming-display-ie true** The default setting is **false**.

Property name	Description
add-outgoing-displaytext-ie {true | false}	*Secondary property.* Specifies whether AA-SBC maps the received displayname in the SIP INVITE From header to the Q931 DisplayTextIE field in outgoing SETUP message. If set to **true**, AA-SBC performs the mapping. **Example: set add-outgoing-displaytext-ie true** The default setting is **false**.
use-server-connection {true | false}	*Secondary property.* Specifies whether AA-SBC creates a new or existing TCP connection. If **true**, AA-SBC uses a TCP connection created by the remote gateway instead of creating a new outbound TCP connection. Use this property for a remote H323 gateway using connection sharing for its H225 traffic. (It uses a single TCP connection for multiple calls.) **Example: set use-server-connection false** The default setting is **true**.

h323-to-sip-fromheader-spec

Purpose

Specifies how to generate a SIP From header from an H.323 SETUP message. When AA-SBC receives a message from an H.323 server via the server that contains this configuration object, it creates the From header using the parameters of this object. The From header is made up of four components defined here--scheme:user@host.suffix.

Syntax

```
config vsp enterprise servers h323-server name
   h323-to-sip-fromheader-spec
```

Properties

Property name	Description
scheme {sip | tel | omit | *string*}	Specifies the Scheme to use in the From (or To) header. Set the system to: • sip—use the SIP scheme. • tel—use the tel scheme. • omit—leaves the field blank. Select this if the upstream server uses the correct scheme in the H.323 SETUP message and you do not want that value changed. • *string*—enters the specified string in the scheme field. **Example: set scheme omit** The default setting is **sip**.
user {calling-number | h323-id | url-id | email-id | omit | *string*}	Specifies the origin of the User field content to use in the From (or To) header. Set the system to derive the information from: • calling-number—the value in the originating phone number. • h323-id—the value from the incoming h323ID alias. • url-id—the value from the incoming url ID. • email-id—the value from the incoming email ID. • omit—leaves the field blank. • *string*—enters the specified string in the User field. **Example: set user h323-id** The default setting is **calling-number**.

Property name	Description
host {h323-id | url-id | email-id | h323gw-domain | omit | *string*}	Specifies the origin of the Host field content to use in the From (or To) header. Set the system to derive the information from. • h323-id—the value from the incoming h323ID alias. • url-id—the value from the incoming url ID. • email-id—the value from the incoming email ID. • h323gw-domain—omits the value configured for the H323 gateway. • omit—leaves the field blank. • *string*—enters the specified string in the User field. **Example: set host h323-id** The default setting is **h323gw-domain**.
suffix {omit | *string*}	Specifies the suffix to add to the From (or To) header. Enter a suffix or select omit to let the system derive the field from the SETUP message. **Example: set suffix omit** The default setting is **omit**.

h323-to-sip-toheader-spec

Purpose

Specifies how to generate a SIP To header from an H.323 SETUP message. When AA-SBC receives a message from an H.323 server via the server that contains this configuration object, it creates the To header using the parameters of this object. The To header is made up of four components defined here—scheme:user@host.suffix.

Syntax

```
config vsp enterprise servers h323-server name
   h323-to-sip-toheader-spec
```

Properties

See h323-to-sip-fromheader-spec for property descriptions.

h323-reason-translate

Purpose

Maps H.323 reason codes to SIP response values. Any call that encounters one of the selected H.323 reasons in response to a call to or from the gateway, returns the configured SIP response value to the SIP process.

Syntax

```
config vsp enterprise servers h323-server name h323-reason-translate
```

Properties

Property name	Description
translation *h323Reason SIPresponse*	Specifies the H.323 reason and the SIP response value. Select a predefined H.323 code and enter a SIP response value. **Example: set translation gkunreachable 404** There is no default setting.

h225-settings

Purpose

The H.225 settings configuration object allows you to configure H.225 on the AA-SBC.

Syntax

```
config vsp enterprise servers h323-server h225-settings
```

Properties

Property name	Description
fast-start [enabled | disabled]	If enabled, the AA-SBC accepts inbound H.323 fast start calls and includes fast start in SETUP messages for outbound H.323 calls. The calls fall back to slow if fast start is unsuccessful. **Example: set fast-start disabled** The default setting is **enabled**.
manual-ringback [enabled | disabled]	If enabled, the AA-SBC prohibits remote ringback. When this property is disabled, SIP to H.323 calls attempt to open an audio channel for remote ringback. **Example: set manual-ringback disabled** The default setting is **enabled**.
use-inbound-call-settings [enabled | disabled]	When enabled for an H.323 to H.323 call, the AA-SBC uses inbound H.323 call settings for H.323 outbound calls. **Example: set use-inbound-call-settings enabled** The default setting is **disabled**.
fwd-progress-as-alerting [enabled | disabled]	When enabled, the AA-SBC sends an Alerting message instead of a Progress message. **Example: set fwd-progress-as-alerting enabled** The default setting is **disabled**
default-terminal-type	Identifies the AA-SBC terminal type. This is used for MSD. **Example: set default-terminal-type 75** Min: 0 / Max: 4294967295 The default setting is **60**.
multiple-calls [enabled | disabled]	When enabled, the AA-SBC allows calls to share an H.225 connection. **Example: set multiple-calls enabled** The default setting is **disabled**.

Property name	Description
maintain-connection [enabled | disabled]	When enabled, the AA-SBC keeps an H.225 connection open after calls are cleared. EXAMPLE: **set maintain-connection enabled** The default setting is **disabled**.
conn-idle-timeout	Specifies the maximum lifetime of an idle H.225 connection. A value of 0 indicates an idle connection should never timeout. **Example: set conn-idle-timeout 2500** Min: 300 / Max: 65536 The default setting is **3600**.
h323-user-alias	Specifies the source and destination address type in Setup, Alerting, Connect, ARQ, and LRQ messages. The following are valid h323-user-alias values: -none -dialedDigits -h323ID -urlID -emailID **Example: set h323-user-alias dialedDigits** The default setting is **none**.
call-alerting-timeout	The maximum number in seconds the AA-SBC waits for Alerting message after sending a SETUP. The call clears if this timeout is reached. **Example: set call-alerting-timeout 10** Min: 0 / Max: 4294967295 The default setting is **4**.
call-establishment-timeout	The maximum number in seconds the AA-SBC waits for an H.323 call to be established. The call clears if this timeout is reached. **Example: set call-establishment-timeout 75** Min: 0 / Max: 4294967295 The default setting is **60**.

Property name	Description
end-session-timeout	The maximum number of seconds the AA-SBC waits after sending a ReleaseComplete before call resources are reclaimed. **Example: set end-session-timeout 30** Min: 0 / Max: 4294967295 The default setting is **15**.
h245-establish-timeout	The maximum number, in seconds, the AA-SBC waits for an H245 connection to be established. The call clears if this timeout is reached. **Example: set h245-establish-timeout 5** Min: 0 / Max: 4294967295 The default setting is **1**.
reinvite-type	*Secondary property.* Indicates if the AA-SBC should use Terminal Capability Set or Extended Fast Connect messages to reconfigure media channels. **Example: set reinvite-type extendedFastConnect** The default setting is **emptyTermCapSet**.
use-progress-inband [enabled | disabled]	When enabled, inband ring information from the inbound H.323 call-leg is propagated to the outbound call-leg. **Example: set use-progress-inband disabled** The default setting is **enabled**.
fwd-retrieve-no-tx [true | false]	When true, the AA-SBC does not pause remotetransmitted if media information is 0.0.0.0. **Example: set fwd-retrieve-no-tx false** The default setting is **true**.
use-server-connection [true | false]	*Secondary property.* specifies whether the AA-SBC creates a new, or uses an existing, TCP connection. If true, the AA-SBC uses a TCP connection created by the remote gateway instead of creating a new outbound TCP connection. Use this property for a remote H.323 gateway using connection sharing for its H.225 traffic. (It uses a single TCP connection for multiple calls.) **Example: set use-server-connection false** The default setting is **true**.

Property name	Description
enum-lookup-called-party [enabled | disabled]	When enabled, the AA-SBC performs an ENUM lookup of the called number before making an outbound H.323 call. **Example: set enum-lookup-called-party enabled** The default setting is **disabled**.
enum-domain	The domain used for ENUM lookups. **Example: set enum-domain 12025551234** The default setting is **e164.arpa**.

h245-settings

Purpose

The H.245 settings configuration object allows you to configure H.245 on the AA-SBC.

Syntax

```
config vsp enterprise servers h323-server h245-settings
```

Properties

Property name	Description
h245-tunnel [enabled | disabled]	When enabled, the AA-SBC attempts to use an H.225.0 connection for H.245 traffic. The use of H.245 tunneling depends on indication from both H.323 terminals and gateways. **Example: set h245-tunnel disabled** The default setting is **enabled**.
early-h245	The AA-SBC does not support early H.245. The following are valid early-h245 values: notunnel—The AA-SBC ignores the early H.245 and completes the call setup using slowstart. reject—The AA-SBC rejects the call. **Example: set early-h245 reject** The default setting is **notunnel**.
wait-for-remote-tcs [true | false]	When true, the AA-SBC waits to receive a Terminal Capability Set message before advertising its capabilities. When false, the AA-SBC issues a TCS message after a slowstart call is connected. **Example: set wait-for-remote-tcs false** The default setting is **true**.
clc-when-pausing-remote [true | false]	*Secondary property.* When true, the AA-SBC closes its TX channels when pausing the remote H.323 terminal. **Example: set cls-when-pausing-remote true** The default setting is **false**.
send-msd-when-unpausing-remote [true | false]	*Secondary property.* Specifies whether the AA-SBC will conduct MSD when using TCS to unpause a remote H.323 gateway. **Example: set send-msd-when-unpausing-remote true** The default setting is **false**.

Property name	Description
use-h450-hold-retrieve [enabled | disabled]	When enabled, the AA-SBC uses H.450 supplemental service PDUs for holds and retrieves. **Example: set use-h450-hold-retrieve disabled** The default is **enabled**.
sip-h323-dtmf-translate <sip-dtmf-type> <h323-dtmf-type>	Sets preferences for H.323-SIP DTMF interworking for a particular H.323 trunk. **Example: set sip-h323-dtmf-translate RFC2833 H245SIGNAL** The default setting is **inband**.
codec-selection	*Secondary property.* Indicates how the AA-SBC chooses converged codecs. The following are valid values: -none—No codec is being used. -local—Use the highest preference common codec seen in SIP SDP. -remote—Use the highest preference common codec in remote TCS. -followMSD—Use the result of MDT to decide. **Example: set codec-selection local** The default setting is **remote**.
map-ptime-to-fpp [true | false]	*Secondary property.* When set to **true**, the AA-SBC uses SDP ptime parameter to set max-frames-per-packet codec value in Terminal Capability Set. Ptime and FPP are not equivalent, however, this allows compatibility in some IW scenarios. **Example: set map-ptime-to-fpp true** The default setting is **false**.

Property name	Description
map-fpp-to-ptime [true | false]	*Secondary property.* When true, the AA-SBC uses max-frames-per-packet codec value in Terminal Capability Set to set SDP ptime parameter. Ptime and FPP are not equivalent, however, this allows compatibility in some interworking scenarios. **Example: set map-fpp-to-ptime true** The default setting is **false**.
add-equivalent-codecs [true | false]	*Secondary property.* When true, the AA-SBC adds equivalent codecs to Terminal Capability Set. The currently supported case is G729 present in SDP which would add both G729 and G729A in TCS. **Example: set add-equivalent-codecs true** The default setting is **false**.

q931-cause-sip-response-map

Purpose

Allows the configuration of q931-cause and/or H.225 reason code for calls cleared by an external SIP UA. When an IW call is cleared on the SIP side, the SIP response code is used to consult an internal table for Q.931/H.225 information needed when generating the ReleaseComplete, Admission Reject, or Location Reject message. By adding a **q931-cause-sip-response-map** entry, you can override the internal table defaults.

Syntax

```
config vsp enterprise servers h323-server q931-cause-sip-response-map
```

Properties

Property name	Description
q931-cause	Select a q931-cause to use when clearing the H.323 side of the call. If this map entry will not generate a q931-cause, or you want to use the default, select **Any**. Example: set q931-cause userbusy
h2250-reason	Select a h225-reason to use when clearing the H.323 side of the call. The following are valid values: • none—use the default h225-reason. • lrj—select lrj if you are generating LRJ messages and enter a relevant reason. • arj—select arj if you are generating ARJ messages and enter a relevant reason. • any—specifying only the q931-cause in this entry. Example: set h225-reason arj The default setting is **none**.
sip-response	Select the sip-response match criteria for this entry. If this entry will not generate a q931-cause or you want to use the default, select **Any**. Example: set sip-response 500 Min: 300 / Max: 699 The default setting is **0**.

q931-settings

Purpose

The q931-settings configuration object is used to configure Q.931 settings on the AA-SBC.

Syntax

```
config vsp enterprise servers h323-server q931-settings
```

Properties

Property name	Description
numbering-plan	The Q.931 numbering plan set in Calling and Called Party number information elements. The following is a list of valid numbering-plan values: -unknown -ISDN -data -telex -national-standard -private -reserved **Example: set numbering-plan private** The default setting is **ISDN**.
numbering-type	The Q.931 numbering type set in Calling and Called Party number information elements. The following is a list of valid numbering-type values: -unknown -international -national -network-specific -subscriber -abbreviated -reserved **Example: set numbering-type national** The default setting is **unknown**.

Property name	Description
presentation-indicator	Enter the static presentation value to use. The following is a list of valid presentation-indicator values: -allowed -restricted -numberNotAvailable -reserved **Example: set presentation-indicator restricted** The default setting is **allowed**.
screening-indicator	Enter the static screening value to use. The following is a list of valid screening-indicator values: -notScreened -verifiedPassed -verifiedFailed -networkProvided **Example: set screening-indicator verifiedPassed** The default setting is **notScreened**.
privacy-dynamic [true | false]	When true, the screening and presentation are dynamic. **Example: set privacy-dynamic false** The default setting is **true**.
use-incoming-display-ie [true | false]	When true, the AA-SBC attempts to use Display IE from the SETUP message when building SIP From: header display-name. **Example: set use-incoming-display-ie false** The default setting is **true**.

Property name	Description
add-outgoing-displaytext-ie [true | false]	When true, the AA-SBC attempts to use SIP From: header display-name when building Display IE in the outgoing SETUP message. **Example: set add-outgoing-displaytext-ie true** The default setting is **false**.
q931-bearer-capability-ie [coding-standard] [info-transfer-capability] [transfer-mode] [transfer-rate] [user-info-layer1-protocol]	Set the Q931. Bearer Capability values used in outgoing H.323 messages. Example: set q931-bearer-capability-ie nationalStd speech packetMode 128Kbps h261

h323-service-routing

Purpose

Configure the H.323 service routing metrics.

Syntax

```
config vsp services-routing h323-service-routing
```

Properties

Property name	Description
metric1 [type] [loadShareScheme]	Sets the metric1 load type. The following are valid values: -none -user-metric -intf-thruput **Example: set metric1 none** The default setting is **user-metric**.
metric2 [type] [loadShareScheme]	Sets the metric2 load type. The following are valid values: -none -user-metric -intf-thruput **Example: set metric2 user-metric** The default setting is **none**.
metric3 [type] [loadShareScheme]	Sets the metric3 load type. The following are valid values: -none -user-metric -intf-thruput **Example: set metric3 user-metric** The default setting is **none**.

Property name	Description
metric4 [type] [loadShareScheme]	Sets the metric4 load type. The following are valid values: -none -user-metric -intf-thruput **Example: set metric4 user-metric** The default setting is **none**.
metric5 [type] [loadShareScheme]	Sets the metric5 load type. The following are valid values: -none -user-metric -intf-thruput **Example: set metric5 user-metric** The default setting is **none**.

federator

Purpose

Secondary object. Creates a federation identity. A federation connects one enterprise server deployment to another in a different organization. Each federated partner communicates with an access proxy (a gateway between internal users and trusted remote users), through which all SIP traffic crossing the network boundary is routed. When configured, this URL is the identity that AA-SBC asserts as the source of traffic when entering a federated domain.

Syntax

```
config vsp enterprise servers server name federator
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables the server participation in a federation. **Example:** set admin enabled The default setting is **disabled**.
url [*user*] [*host*] [*display*] [*param*] [*port*] [any | UDP |TCP | TLS]	Sets the user information for access to federated domains. The system needs this information when configured for IP phone-to-server client calling capability. In essence, the federator is the URL used as the source for the call into the server. Enter the following: • **user**—user name • **host**—a domain name or an IP address • **display**—display name • **param**—a user parameter, for example user=phone or user=ip • **port**—the local port the system uses to communicate with the server. • **transport**—the protocol the system uses to communicate with the server. **Example: set url jdoe@abc.com** There is no default setting for any option but transport. The default transport setting is **any**.

default-sip-settings

Purpose

Secondary object. Configures SIP communications settings for calls destined for the server. These settings override SIP settings from the default session configuration. However, these settings are overridden from outside the session configuration by SIP settings contained in any matching policy rules.

Syntax

```
config vsp enterprise servers server name default-sip-settings
```

Properties

The properties of this object are the same as those for sip-settings. See that object for property descriptions.

server-pool

Purpose

Creates a configuration of servers that AA-SBC uses to access enterprise information. A server pool is a logical construct used to group physical interfaces (hosts) into a shared resource for registration and INVITE requests. Add servers to the pool using the server-pool-admission-control subobject.

Syntax

```
config vsp enterprise servers server name server-pool
```

Properties

Property name	Description
call-routing-on {request-uri | to-uri | as-is}	*Secondary property.* Specifies whether the system does routing or location lookups based on the Request URI, the To URI, or an alternate setting. By default, the system performs lookups on the Request URI. Change this setting, for example, when routing information is not available in the Request URI but it is available in the To URI. This setting applies to all servers in the pool. All calls from all servers in the pool are looked -up based on the URI set with this property. This setting can also be configured in the arbiter object. If values are set in both this and the arbiter, the arbiter settings take precedence. The system does the lookup on either: • **request-URI**—the Request URI, which contains the hop-by-hop destination for the call. • **to-uri**— the To URI, which contains the final destination of the call. • **as-is**—the Request URI (the default) or the value set for this property in the arbiter object. **Example:** set call-routing-on to-uri The default setting is **request-uri**.

Property name	Description
handle-response *code* {try-next-peer | try-next-route | forward}	Specifies the action the system should take when it receives a specific response code from this server. Enter a code, and set a handling pattern: • **try-next-peer**—the system forwards the message to the next server within this server pool. • **try-next-route**—the system forwards the message to the route that is the next most-specific. Use this in conjunction with the **arbiter-apply** joined-matches option (in the arbiter object). • **forward**—the system returns the response to the originator of the message. **Example: set handle-response 404 try-next-route** The default setting for the handling pattern is **try-next-peer**.

Property name	Description
dialog-failover	When enabled, the **dialog-failover** setting forces AA-SBC to check the state of the destination SIP server before sending messages. If the destination server is down, the calls are routed to the next configured (and available) backup server. **Note:** For dialog-failover to work, the **failure-detection** property must be set to *auto*, *ping*, or *register* in the **servers** and/or **exchange** objects. When **dialog-failover** is set to disabled, any calls in progress at the time of the failure will be retried at the original destination server until the configured timeout settings have expired. **Example: set dialog-failover enabled** The default setting is **disabled**.
server-gatekeeper-id	*Secondary property*. Specifies the way the AA-SBC reaches an H.323 Gatekeeper. dynamic—The AA-SBC learns the Gatekeeper ID via RAS messaging. static—The GKId string must be configured. The AA-SBC uses this configured string to contact a remote H.323 Gatekeeper. **Example**: **set server-gatekeeper-id dynamic** The default value is **dynamic**.

server-pool-admission-control

Purpose

Allows you to configure a server-pool CAC on any enterprise server that contains a pool.

Syntax

```
config vsp enterprise servers server name server-pool
   server-pool-admission-control
```

Properties

Property name	Description
max-bandwidth	Enter the maximum amount of bandwidth, in kbits per second, the AA-SBC allocates to the AOR. When the system reaches the maximum bandwidth limit for a server, it rejects calls until bandwidth use drops below the maximum. **Example**: set max-bandwidth 10000 Min: 0 / Max: unlimited The default setting is **unlimited**.
max-number-of-concurrent-calls	Specify the maximum number of active calls allowed for this AOR at one time. When this value is reached, the connection does not accept calls until the value drops below the threshold. **Example**: **set max-number-of-concurrent-calls 5000** Min: 0 / Max: 1000000 The default setting is **1000**.
max-calls-in-setup	Sets the maximum number of simultaneous call legs in setup stage that are allowed for this AOR. A call leg in setup is much more compute-intensive than established call legs, so this value is more restrictive than the concurrent call leg value. A value of 0 causes the system to decline all calls and registrations. **Example**: **set max-calls-in-setup 5000** Min: 0 / Max: 10000 The default setting is **30**.

Property name	Description
call-rate-limiting	Limits the number of calls sent to an AOR within a certain interval in seconds. Once this interval is reached, the system rejects any calls to or from this AOR until the rate decreases, returning a response code and message. This feature sets the acceptable arrival rate for incoming calls when used with admission-control and the acceptable set-up rate when used with emission-control. When this feature is enabled, set the number of calls and the measurement interval. You can also enter a result code from 400 to 699 and a text string to accompany call rejection if no available server is found. **Example**: **set call-rate-limiting enabled** The default setting is **disabled**
admission-control {enabled | disabled}	Specifies whether the system considers AOR limitations when forwarding a call from the AOR. The system tracks the number of concurrent (both incoming and outgoing) active calls for this AOR. **Example**: **set admission-control enabled** The default setting is **disabled**.
emission-control	Specifies whether the system considers AOR limitations when forwarding a call to this AOR. The system tracks the number of concurrent (both incoming and outgoing) active calls for this AOR. **Example**: **set emission-control enabled** The default setting is **disabled**.

Property name	Description
call-admission-control-error-code	Enter the call admission error code. **Example**: **set call-admission-control-error-code 700** Min: 400 / Max: 999 The default setting is **503**.
call-admission-control-error-string	Enter the text string the users sees when a call admission control error occurs. **Example**: **set call-admission-control-string cac error**
call-emission-control-error-code	Enter the call emission error code. **Example**: **set call-emission-control-error-code 800** Min: 400 / Max: 999 The default setting is **503**.
call-emission-control-error-string	Enter the text string the user sees when a call emission control error occurs. **Example**: **set call-emission-control-error-string cec error**

server

Purpose

Adds server connections to the server pool. You must identify the server by host name or IP address. Optionally, you can set a preference, protocol, port, load-balancing, and other criteria, as well as limit connections.

Each server in a pool has an associated order of preference. The server with the lowest order is preferred, and becomes the primary. The other servers are backups. If the primary is up, as indicated by the AA-SBC monitoring process, then registrations and INVITE requests are sent there. If it is down (but the backup is up), then AA-SBC sends requests to the backup. When the primary later becomes available again, AA-SBC resumes sending registrations and INVITE to it.

If both the primary and backup are down, then the peer changes to local mode. In local mode, AA-SBC does not proxy registrations to the failing peer. Instead, if the To header is addressed to a SIP phone directly connected to AA-SBC (it has a location binding with AA-SBC), INVITEs are switched locally. If the To header is addressed to a location out of reach of AA-SBC, the INVITE is forwarded to the configured PSTN gateway: (The gateway is configured with the **sip-gateway** server object.)

If the primary or backup later becomes available, the peer reverts from local back to provider mode, and again forwards registrations and INVITE to the primary or backup server.

The server object allows you to configure normalization plans for outgoing and incoming calls. See Chapter 46, "Outbound and inbound normalization objects", for a full description of the server object outbound-normalization and inbound-normalization subobjects.

Syntax

```
config vsp enterprise servers server name server-pool server name
```

Properties

Property name	Description
host *server*	Specifies the host name or IP address of an Internet endpoint. Enter a host name or IP address. **Example: set host 192.168.10.10** There is no default setting.
endpoint *string*	Associates a text string with a server-pool server. The string can later be used to group and categorize servers. **Example: set endpoint server1** There is no default setting.
transport {any | UDP | TCP | TLS}	Specifies the protocol used by the connection. **Example: set transport any** The default protocol is **UDP**.
port *portNumber*	Specifies the port used by the connection for SIP traffic. **Example: set port 3333** The default setting is port **5060**.

Property name	Description
local-port *portNumber*	Sets a port number for the system to use in the Contact header, Via header, and source port when it sends a Register request (and subsequent SIP messages) to an upstream server. The server caches the binding and includes the local-port when contacting the system. Additionally, the server can be configured to send SIP messages to this particular local-port without prior registration from the system. With local-port configured, the system can tell: • to which connection in the server pool to forward a call. • which connection in the server pool it received the call from, when the connection sends SIP message to this local port. Using this property allows you to group traffic based on the local port number. For example, if there are multiple domains from a single physical server, the port will indicate which domain should receive the call. Or, if there is a distinct pair of physical servers to protect traffic for a domain, the Eclipse can fail over to the right backup server (in case of primary failure) for this particular domain. **Example: set local-port 50501** There is no default setting.
connection-role {initiator | responder}	Specifies the way the server behaves in establishing a TCP/TLS connection. If set to **initiator**, the server can open up a connection without any SIP traffic. If set to **responder**, the server will not open up a TCP/TLS connection until receiving SIP traffic. **Example: set connection-role responder** The default setting is **initiator** for a server-pool server and responder for a sip-connection.

Property name	Description
connection-retry-interval *seconds*	Specifies the number of seconds the system waits between attempts to open a TCP or TLS connection. This value is only meaningful if the **connection-role** property is set to **initiator**. If set to **responder**, the value is ignored. **Example: set connection-retry-interval 10** The default setting is **5** seconds.
preference {none | *preference*}	Specifies the preference for the connection. The lower the value the higher the preference. If you use the value of **none**, the system uses the preference set in a different part of the configuration, such as the ordered set of arbitration rules in the dial-plan object. **Example: set preference 1** The default preference setting is **none**.
admission-control {enabled | disabled}	Specifies whether the system considers downstream server capacity when forwarding a call from the server. AA-SBC tracks the number of concurrent calls for each server. If this property is **enabled**, the system does not forward calls from the server if the server limit has been reached and instead sends a “503 Service Unavailable” message. If **disabled**, the system does forward calls from the server. (Set the call limit with the **max-number-of-concurrent-calls** property.) See Admission control for an AOR for specific information on CAC settings applicability for an AOR. **Example: set admission-control enabled** The default setting is **disabled**.

Property name	Description
emission-control {enabled | disabled}	Specifies whether the system considers upstream server capacity when forwarding a call to the server. AA-SBC tracks the number of concurrent (both incoming and outgoing) active calls for the server. If this property is **enabled**, the system does not forward calls to the server if the limit, set with the **max-number-of-concurrent-calls** property, has been reached. Instead, the system sends one of the following messages and drops the call: • If there is one outbound server/UAC/UAS, the system sends a "486 Busy" message, indicating that the route was resolved but that the AOR was unavailable. • If there are multiple outbound server/UAC/UASs and all have reached the maximum concurrent calls threshold, the system sends a "486 Busy" message. • If there are multiple outbound server/UAC/UASs and at least one has not reached the maximum concurrent calls threshold, the return code is determined by the final server that the system attempted to reach. This could be, for example, "486 busy" or a "504 server timeout" if the last server was unresponsive and the transaction timed out. If **disabled**, the system continues to forward calls to the server. **Example: set emission-control enabled** The default setting is **disabled**.

Property name	Description
max-bandwidth {unlimited | *kbps*}	Specifies the amount of bandwidth the system allocates to a connection. For a SIP server, the default value is **unlimited** or the server uplink bandwidth. For example, if the uplink is GigE, then bandwidth is 1 million kbps. When the system reaches the maximum bandwidth limit for a server, it rejects calls until bandwidth use drops below the maximum. Note that the bandwidth usage value is based not on the actual traffic on the wire, but on a calculation done by the system. The calculation uses the value associated with the first known CODEC identified in the SDP for a usage rate. If there is not a known CODEC, or the value has not yet been determined from the SDP, the system uses the **default-session-bandwidth** value from the session configuration media object. Set a specific bandwidth if you are using, for example, a TDM trunk or PSTN gateway with limited bandwidth. For a PSTN trunk, the usual capacity is DS0 (64 kbps bandwidth). If a gateway has 8 trunks, then the gateway has 512 kbps bandwidth. **Example: set max-bandwidth 512** The default setting is **unlimited**.
max-number-of-concurrent-calls *integer*	Specifies the number of calls allowed on the connection at one time. When this value is reached, the connection will not accept calls until the value drops below the threshold. **Example: set max-number-of-concurrent-calls 1500** Enter a value between 0 and 1,000,000; the default is **1000** calls. A value of 0 causes the system to decline all calls and registrations.

Property name	Description
max-calls-in-setup *integer*	Sets the maximum number of simultaneous inbound and outbound call legs in setup stage that are allowed for the connection. A call leg in setup is much more compute-intensive than established call legs, so this value is more restrictive than the concurrent call leg value. A value of 0 causes the system to decline all calls and registrations. **Example: set max-calls-in-setup 50** Enter a value between 0 and 10,000; the default is **30** call legs.
call-rate-limiting {enabled [*callsPerInterval*] [*interval*] [*resultCode*] [*resultString*] | disabled}	*Secondary property.* Limits the number of calls sent to a server within a certain interval. Once this interval is reached, the system hunts for the next available server. If there are no available servers, the system returns a response code and message. This feature sets the acceptable arrival rate for incoming calls when used with **admission-control** and the acceptable set-up rate when used with **emission-control**. If **enabled**, set the number of calls allowed and the measurement interval (in seconds). You can also enter a result code from 400 to 699 and a text string to accompany call rejection if no available server is found. **Example: set call-rate-limiting enabled 50 1 480 “Temporarily unavailable”** The default value is **disabled**. If set to **enabled**, the default calls-per-interval is 60, the default interval is 1 second, and the default result is 486, Busy Here.

Property name	**Description**
max-number-of-registrations *value*	*Secondary property.* Specifies the maximum number of registrations that can be active with a server at any one time. This property is used in conjunction with the **server-registration-balance** property of the VSP settings object to implement registration load balancing. **Example: set max-number-of-registrations 1500** Enter a value between 0 and 1,000,000; the default setting is **1000** registrations. A value of 0 causes the system to decline all calls and registrations.
max-registrations-in-progress *value*	*Secondary property.* Specifies the number of registrations or authentication requests per second that the system forwards to the server. Use this property as a flow control mechanism to control the system, which can process registrations much more quickly than the server. To set this, you must know the capability of your server. You also must enable the **server-registration-balance** property of the VSP settings object. When a register is delegated/forwarded/tunneled to the server, the system increments a cluster-wide server counter. When the counter reaches this threshold, the system handles subsequent registrations. It responds with “200 OK,” but sets a brief expiration, causing the phone to reregister almost immediately. **Example: set max-registrations-in-progress 600** Enter a value between 0 and 10,000; the default setting is **300** registrations. A value of 0 causes the system to decline all calls and registrations.
external-outbound-normalization {no | yes *server*}	*Secondary property.* Specifies whether the system should perform external normalization on outbound call legs. Enter the host name of your calling plan server. **Example: set external-outbound-normalization yes ITALKBB-EGR5** The default setting is **no**.

Property name	Description
external-inbound-normalization {no | yes *server*}	*Secondary property.* Specifies whether the system should perform external normalization on inbound call legs. Enter the host name of your calling plan server. **Example: set external-inbound-normalization yes PT1-INGRESS** The default setting is **no**.
handle-unregister-locally {enabled | disabled}	*Secondary property.* Specifies under what circumstances AA-SBC can process an UNREGISTER request. When **enabled**, the system processes the request even if the REGISTER was not sent by the same registration endpoint. **Example: set handle-unregister-locally enabled** The default setting is **disabled**.

error-response-codes

Purpose

Opens the error-response-codes configuration object where you can specify custom admission-control, emission-control, and server down error codes and text strings that will be returned in the SIP response and captured in the event log.

Syntax

```
config vsp enterprise servers type name server-pool server name
   error-response-codes
```

Properties

Property name	Description
call-admission-control-error-code	Modifies the error code to be returned in the SIP response when a call-admission-control error occurs. **Example**: **set call-admission-control-error-code 410** The default setting is **503** (Service Unavailable)
call-admission-control-error-string	Creates a user-defined text string to be returned in the SIP response when a call-admission-control error occurs. **Example: set call-admission-control-error-string this is an error**
call-emission-control-error-code	Modifies the error code to be returned in the SIP response when a call-emission-control error occurs **Example**: **set call-emission-control-error-code 410** The default setting is **503** (Service Unavailable)
call-emission-control-error-string	Creates a user-defined text string to be returned in the SIP response when a call-emission-control error occurs. **Example**: **set call-emission-control-error-string this is an error**
server-down-error-code	Modifies the error code to be returned in the SIP response when a destination SIP server is down. **Example: set server-down-error-code 403** The default setting is **502** (Bad Gateway)
server-down-error-string	Creates a user-defined text string to be returned in the SIP response when a destination SIP server is down. **Example: server-down-error-string error**

registration-proxy

Purpose

Sets the characteristics of the relationship between two peers that are both proxies. When AA-SBC acts as a proxy, it is able to supply the credentials needed for authentication challenges. It maintains a location service database to store SIP caller location (address-of-record) information. This database can be updated via AA-SBC registration service, static address-of-records (AORs), and/or configured AA-SBC policies. To ensure that peer systems have and use the same database, set the properties of this object.

You can also set the registration-proxy property in the vsp object. In that instance, you are optimizing system performance by specifying whether AA-SBC should walk the database.

Syntax

```
config vsp enterprise servers server name registration-proxy
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables this proxy registration configuration. If **enabled**, the server applies these characteristics to sessions with its configured peers. If **disabled**, these characteristics are inactive. **Example: set admin disabled** The default setting is **enabled**.
request-download {no | yes [*minutes*]}	Automates the download of the registration database from a peer. (Peers are identified in the server configuration.) If set to **yes**, the peer system downloads the database to this server with the frequency set in the request interval of this property. In addition, it copies the interval to the expiration time in the REGISTER requests forwarded to peers. If set to **no**, downloads do not occur automatically. The system only learns of new or changed AORs through REGISTER requests. **Example: set request-download yes 1080** The default setting is **no**. If set to **yes**, the default interval is 1440 minutes (24 hours).

network

Purpose

Sets the properties specific to the server socket. To set general system network parameters, including other socket properties, use the services network object.

Only applicable to

- `sip-connection`

Syntax

```
config vsp enterprise servers sip-connection name network
```

Properties

Property name	Description
`tcp-keepalive-time` *`seconds`*	Specifies the time, in seconds, that an established TCP connection can remain idle before the system sends a keepalive to the client. The idle time expiration initiates the keepalive process. **Example: set tcp-keepalive-time 900** Enter a value from 10 to 14,400. The default setting is **600** seconds.
`tcp-keepalive-probes` *`probes`*	Specifies the number of unanswered TCP keepalive probes that are allowed before the system determines a session is idle and disconnects it. **Example: set tcp-keepalive-probes 10** Enter a value from 2 to 16. The default setting is **5** probes.
`tcp-keepalive-interval` *`seconds`*	Specifies the time, in seconds, that the system waits for a response from a keepalive probe before ending the next one. AA-SBC continues to send probes until it has sent the number specified in the **tcp-keepalive-probes** property. **Example: set tcp-keepalive-interval 10** Enter a value from 1 to 60. The default setting is **6** seconds.

ccs

Purpose

Configures AA-SBC to recognize the Avaya Converged Communication Server (CCS), a SIP proxy server that connects to the company's proprietary IP telephony solution, the Avaya Call Manager (ACM). The CCS will add SIP-based voice, presence, and Instant Messaging (IM) services.

This feature is only applicable to Avaya.

Syntax

```
config vsp enterprise servers server name ccs
```

Properties

Property name	Description
server *ipAddress*	Sets IP address of the computer hosting the CCS service. **Example: set server 192.168.10.10** There is no default setting.
port *port*	Sets the TCP port of the SIP proxy service on the computer hosting the CCS service. **Example: set port 2020** There is no default setting.
mode {edge | home | home-edge}	Specifies the type of server this CCS server is functioning as: • **edge**—an edge device, allowing domain routing. • **home**—a home device with only one domain configured. • **home-edge**—a combination device. **Example: set mode home** The default setting is **home-edge**.

60. Services routing objects

Services routing description

The **services-routing** object applies cluster-wide routing configurations, such as metrics to each of the AA-SBC service routing tables (media, SIP, H.323, and STUN) and gateway health checks. Metrics define the cost associated with each route (determine the preference of one route over another), controlling how services are load-balanced across the cluster.

There are four service routing table types:

- Media— used for allocating media resources across the cluster
- SIP—used for routing SIP packets on each AA-SBC.
- H.323—Used for routing H.323 packets on each AA-SBC
- STUN—used for allocating STUN requests

When you configure an IP interface, AA-SBC installs both a network route and a host route into the generic routing table, along with a metric (preference) for the route. If there are services configured under the interface (e.g., media, SIP, or STUN), the route is also installed in the specific service routing table. In addition, each service route table can have subtables that are created when a routing-tag is associated with an **ip** interface. See Tag-based route selection for a description of these tag-based tables.

Understanding the application of route metrics

Each service route entry has five metric fields that serve as tie breakers to determine route preference. You can assign different metric types to each of the five metric fields, controlling the preference of one route over another. Given that there may be multiple paths available to reach a destination, using service route table metrics allows AA-SBC to consider routes differently depending on their purpose. The lower the metric value, the higher the preference.

For each service route table (and therefore, application), you can set different criteria for route selection. For example, to influence media anchoring route selection you can assign metrics to the media table. If two or more routes exist with equivalent metric values, these routes are considered equal-cost routes. AA-SBC uses round-robin through all equal-cost routes when distributing a service across a cluster. You can assign metric types to a service route table using the five metric fields to prefer one route over another.

To select the most preferred route(s), AA-SBC applies these table-specific metrics. If two or more routes exist to reach a destination, AA-SBC first compares metric1. If metric1 has the same value for the routes, AA-SBC then compares metric2, and so on, up to metric5, or until a route is preferred based on a metric.If all five metrics are equivalent, the routes are considered equal cost.

By default, AA-SBC uses the **user-metric**. This value is used as the first tiebreaker (metric1 field), and the remaining fields default to **none**.

The frequency with which AA-SBC updates the value for calculated metrics (for all service metric tables) is set with the VSP **services-routing** object **metric-timer** property. You select the metric type for a service with the **media-service-routing**, **sip-service-routing**, and **stun-service-routing** objects. Note that any configuration changes to the assignment of metric types causes an immediate recalculation of that service's route table. Use the **show services-routing-metrics** command to display the metric assignments.

The following lists and describes each metric type.

- none—AA-SBC does not assign any metric value. If all metrics are set to none, multiple routes to reach a destination have the same preference. AA-SBC chooses a route using the round robin algorithm.
- user-metric—A user-defined static metric that can be associated wiht a specific IP interface or static route. This static metric can be used to determine the preference of a route. The user-metric value can be thought of as the cost associated with a route, the lower the cost the more preferred the route.

  For an IP interface this metric value is configured under **box** > **interface** > **ip** > **metric**. For a static route this metric value is configured under **box** > **interface** > **ip** > **routing** > **route** > **metric**.

- intf-thruput—A route that is associated with an IP interface. This metric type uses the dynamic throughput, in kilobits per second, of the physical interface that a particular route is associated with to determine route preference. The lower the intf-thruput value, the more preferred the route.
- box-cpu-load—A route that is associated with a AA-SBC. This metric type uses the AA-SBC-wide CPU load to determine route preference. The lower a AA-SBC's CPU load is, the more preferred the route.
- box-memory—A route associated with a AA-SBC. This metric type indicates teh percentage of memory allocated from the SIP Process's maximum heap. The lower the box-memory, the more preferred the route.
- box-media-load—This metric type is currently unused.

About Services Routing Load Balancing

On the AA-SBC, services routing distributes the load of a service using either a round-robin (RR) or a weighted-round-robin (WRR) load balancing algorithm. The RR algorithm loops through a set of equal cost routes, distributing the load equally across the set of routes. When dynamic metrics such as a AA-SBC's CPU utilization is used to balance the load, the RR algorithm can lead to an uneven distribution of the load. The WRR load balancing algorithm is a better choice when using dynamic metrics such as CPU utilization.

For each type of service routing, there are five service routing metric fields named metric1 through metric5. Each of the metrics can be configured to use a specific metric type to determine the services route preference.

You configure each metric to determine which load balancing algorithm, either **round-robin** or **weighted-round-robin**, is used using the **load-share-scheme** property. The default algorithm is **round-robin**.

About RR Behavior on the AA-SBC

When all metric fields are configured for RR, the AA-SBC determines which routes are part of an equal cost route set. If all metric fields configured for RR are of equal value for two or more routes, those routes are considered equal cost. In the example below, Route 1 and Route 3 are considered equal cost because metric1 and metric2 are configured as RR and both metrics have the same values. Route 2 would be a secondary route because a metric 2 value of 20 is less-preferred than Route 1 and Route 3.

Route Name	Metric1 — user-metric RR	Metric2 — box-cpu-load RR
Route 1	1	10
Route 2	1	20
Route 3	1	10

About WRR Behavior on the AA-SBC

If one or more metrics are configured as WWR, then those metrics are used to calculate the load-share for each route in an equal cost set. Each metric configured for WRR has a dynamic weight that is calculated based on how close a metric is to its upper bound. A route’s weight is then used to calculate its load share across an equal cost route set. Services routing then balances the load across the equal cost route set based on the load share values calculated for each route. In the example below, Routes 1-3 are considered equal cost because their round robin metrics are all of equal value. Metric2 is configured to use box-cpu-load (% of CPU utilized) as a weighted-round-robin. Based on these WRR values, the routes are assigned a load-share. Since Route 1 is the least loaded, it is assigned the highest load-share (75) while route 3 receives the lowest load-share (25). Once the load-share of the equal cost route is calculated, the load is distributed appropriately across the route set. In this example, Route 1 should receive twice as much load as Route 3, and 25% more load than route 2. For example, if 150 requests are made, 75 are distributed using Route 1, 50 by Route 2, and 25 by Route 3. These values are based on the calculated loadshare value.

Route Name	Metric1 — user-metric RR	Metric2 — box-cpu-load RR
Route 1	1	25 (load-share 75)
Route 2	1	50 (load-share 50)
Route 3	1	75 (load-share 25)

The values of a route-s metric fields are updated periodically based on the **vsp > services-routing > metric-time** property. Each time this metric-time expires, the load-shares are recalculated based on the latest metric values. Also, anytime an equal cost route is added or removed from an equal cost route set, new load-share values are calculated.

services-routing

Purpose

Configures VSP routing to control routing in a cluster of one or more AA-SBC devices. For example, you can set the interval with which metrics used for route selection are updated. When this interval expires, AA-SBC recalculates the selected metrics for each table and distributes the value throughout the cluster. At that point, AA-SBC recalculates the preference of each route affected by a metric change. In addition, the subobjects provide access to the metric selection for each of the service routing tables.

Gateway health checks

This object also configures gateway health checks for static routes, verifying reachability of a gateway. (When configuring a static **route**, you must supply a gateway address as the next-hop for forwarding packets to reach the destination network.) AA-SBC sends ARP requests to each configured gateway, from each configured interface using that gateway. If a gateway is configured on multiple interfaces, AA-SBC verifies reachability from each interface.

AA-SBC uses a configurable timer and a maximum failure count to determine when a gateway is considered unreachable. The timer controls how often a health check is performed. The maximum failure count controls how many consecutive health checks must fail before the gateway is considered unreachable. For example, if the gateway health timer is set to 10 seconds, and the gateway maximum failure count is set to 3, a gateway would become unreachable after 3 consecutive health check failures or 30 seconds. When a gateway fails its health check on a particular IP interface, AA-SBC considers any static routes configured with that gateway on that IP interface unreachable.

Syntax

```
config vsp services-routing
```

Properties

Property name	Description
metric-timer *seconds*	Sets the interval (in seconds) with which the system updates service-routing metrics. This interval, which applies to all route service tables, controls how quickly the system can propagate a change in a metric value throughout the cluster service route tables. **Example: set metric-timer 90** Enter a value of 0 through 86400 (1 day); the default setting is **60** seconds. A setting of 0 disables metric updates.
gateway-health-timer *seconds*	Sets the frequency with which the system sends ARP requests to a gateway. This interval applies to all gateways configured on the box. See Gateway Health ChecksGateway Health Checks for more information. **Example: set gateway-health-timer 90** Enter a value of 0 through 86400; the default setting is **0** seconds. A setting of 0 disables the gateway health check feature.
gateway-max-failures *integer*	Specifies the number of consecutive health checks that must fail before the system considers a gateway unreachable. See Gateway Health Checks for more information. **Example: set gateway-max-failures 5** Enter a value of 1 through 1000; the default setting is **3** failures.

Property name	Description
cpu-sample-interval *seconds*	Sets the sampling interval (in seconds) in which to calculate the CPU usage of the system. For example, if the **cpu-sample-interval** is 10 seconds, then 10, one second samples will be used to calculate the CPU usage. The result of this calculation is a CPU usage percentage between 0 and 100. The calculated CPU usage value is used as the **box-cpu-load** metric. **Example: set cpu-sample-interval 90** Enter a value of 1 through 3600; the default setting is **60** seconds.
cpu-sample-divisor *integer*	Normalizes the **box-cpu-load** metric. To do so, the raw CPU usage percentage is divided by this value and the result is used as the **box-cpu-load** metric. This provides better control of how services are load balanced, and results in a better load balancing distribution. For example, if three boxes have a CPU usage of 21, 22, and 23 percent respectively, without normalization of **box-cpu-load** metric, the box with CPU usage of 21 would be preferred over the other two boxes. By normalizing the CPU usage percentage using the default divisor of 10, the metric values for all three boxes becomes two. The load will then be balanced across all three boxes. **Example: set cpu-sample-divisor 20** Enter a value of 1 through 100; the default setting is **10**.

media-service-routing

Purpose

Sets the basis for route evaluation when selecting routes from the media service routing table. AA-SBC uses these metrics, which set route precedence, when anchoring media.

In an AA-SBC cluster, media can be load balanced across two or more media AA-SBCs. The decision as to which AA-SBC handles the media is determined based on the signaling address from the initial INVITE. A services routing lookup is performed in a cluster-wide basis using this signaling address to determine which AA-SBC should handle the media. Once a media AA-SBC is selected, all the media resources for the session are allocated locally from that media AA-SBC.

Delayed-Offer Topology

In a Delayed-offer topology, local media ports must be configured on the signaling AA-SBC. If you want to have the signaling AA-SBC handle only signaling and no media, you can load balance the media across the media AA-SBCs. This is accomplished by configuring each of the signaling interfaces with a higher metric than any of the media interfaces on the two media AA-SBCs. Any static routes configured under these signaling interfaces should also be configured with the same higher metric. This is configured under the **ip > routing > route > metric <cost>** property. By configuring the media interfaces on the media A-SBCs with a lower metric, they are always preferred over the signaling AA-SBCs, allowing all media to load balanced across the media SBCs only.

Syntax

```
config vsp services-routing media-service-routing
```

Properties

Property name	Description
metric1 <metric-type> <load-share-scheme>	Sets the first type of metric considered by the system when it selects a route for media traffic. **Example: set metric1 intf-throughput weighted-round-robin** The default setting is **user-metric** and **round-robin**.
metric2 <metric-type> <load-share-scheme>	Sets the second type of metric considered by the system as criteria for route selection. **Example: set metric2 user-metric weighted-round-robin** The default setting is **none** and **round-robin**.
metric3 <metric-type> <load-share-scheme>	Sets the third type of metric considered by the system as criteria for route selection. **Example: set metric3 box-cpu-load weighted-round-robin** The default setting is **none** and **round-robin**.
metric4 <metric-type> <load-share-scheme>	Sets the fourth type of metric considered by the system as criteria for route selection. **Example: set metric4 box-memory weighted-round-robin** The default setting is **none** and **round-robin**.
metric5 <metric-type> <load-share-scheme>	Sets the last type of metric considered by the system as criteria for route selection. **Example: set metric5 user-metric weighted-round-robin** The default setting is **none** and **round-robin**.

sip-service-routing

Purpose

Sets the basis for route evaluation when selecting routes from the SIP service routing table. AA-SBC uses these metrics, which set route precedence, for SIP signaling. See Understanding the Application of Route Metrics for a description of using metrics for route selection.

Syntax

```
config vsp services-routing sip-service-routing
```

Properties

Property name	Description
metric1 <metric-type> <load-share-scheme>	Sets the first type of metric considered by the system when it selects a route for SIP signaling traffic. **Example: set metric1 intf-throughput weighted-round-robin** The default setting is **user-metric** and **round-robin**.
metric2 <metric-type> <load-share-scheme>	Sets the second type of metric considered by the system as criteria for route selection. **Example: set metric2 user-metric weighted-round-robin** The default setting is **none** and **round-robin**.
metric3 <metric-type> <load-share-scheme>	Sets the third type of metric considered by the system as criteria for route selection. **Example: set metric3 box-cpu-load weighted-round-robin** The default setting is **none** and **round-robin**.

Property name	Description
metric4 <metric-type> <load-share-scheme>	Sets the fourth type of metric considered by the system as criteria for route selection. **Example: set metric4 box-memory weighted-round-robin** The default setting is **none** and **round-robin**.
metric5 <metric-type> <load-share-scheme>	Sets the last type of metric considered by the system as criteria for route selection. **Example: set metric5 user-metric weighted-round-robin** The default setting is **none** and **round-robin**.

h323-service-routing

Purpose

Sets the basis for route evaluation when selecting routes from the H.323 service routing table. AA-SBC uses these metrics, which set route precedence, for H.323 signaling. See Understanding the Application of Route Metrics for a description of using metrics for route selection.

Syntax

```
config vsp services-routing h323-service-routing
```

Properties

Property name	Description
metric1 <metric-type> <load-share-scheme>	Sets the first type of metric considered by the system when it selects a route for SIP signaling traffic. **Example: set metric1 intf-throughput weighted-round-robin** The default setting is **user-metric** and **round-robin**.
metric2 <metric-type> <load-share-scheme>	Sets the second type of metric considered by the system as criteria for route selection. **Example: set metric2 user-metric weighted-round-robin** The default setting is **none** and **round-robin**.
metric3 <metric-type> <load-share-scheme>	Sets the third type of metric considered by the system as criteria for route selection. **Example: set metric3 box-cpu-load weighted-round-robin** The default setting is **none** and **round-robin**.
metric4 <metric-type> <load-share-scheme>	Sets the fourth type of metric considered by the system as criteria for route selection. **Example: set metric4 box-memory weighted-round-robin** The default setting is **none** and **round-robin**.
metric5 <metric-type> <load-share-scheme>	Sets the last type of metric considered by the system as criteria for route selection. **Example: set metric5 user-metric weighted-round-robin** The default setting is **none** and **round-robin**.

stun-service-routing

Purpose

Sets the basis for route evaluation when selecting routes from the STUN service routing table. AA-SBC uses these metrics, which set route precedence, when sending requests to a STUN server, for example. See Understanding the Application of Route Metrics for a description of using metrics for route selection.

Syntax

```
config vsp services-routing stun-service-routing
```

Properties

Property name	Description
metric1 <metric-type> <load-share-scheme>	Sets the first type of metric considered by the system when it selects a route for STUN server traffic. **Example: set metric1 intf-throughput weighted-round-robin** The default setting is **user-metric** and **round-robin**.
metric2 <metric-type> <load-share-scheme>	Sets the second type of metric considered by the system as criteria for route selection. **Example: set metric2 user-metric weighted-round-robin** The default setting is **none** and **round-robin**.
metric3 <metric-type> <load-share-scheme>	Sets the third type of metric considered by the system as criteria for route selection. **Example: set metric3 box-cpu-load weighted-round-robin** The default setting is **none** and **round-robin**.

Property name	Description
metric4 <metric-type> <load-share-scheme>	Sets the fourth type of metric considered by the system as criteria for route selection. **Example: set metric4 box-memory weighted-round-robin** The default setting is **none** and **round-robin**.
metric5 <metric-type> <load-share-scheme>	Sets the last type of metric considered by the system as criteria for route selection. **Example: set metric5 user-metric weighted-round-robin** The default setting is **none** and **round-robin**.

61. Services objects

Services description

The services objects allow you to configure AA-SBC event logging and virus scanning services.

Using filters with event log messages

AA-SBC uses the filters you define in the event-log subobjects to determine the type and severity of event messages to send to the destination target. When creating a filter you specify the log class, which selects which type of event messages to send. Use the question mark at the command line to display the complete list of log classes. After selecting a class, select a severity level.

Event log messages can be written to:

- a remote syslog server (syslog)
- a file (file)
- the local database (local-database)
- an external database (external-database)
- the CLI window (cli)
- an SMTP server (smtp)
- a Tivoli server (tivoli)

When configuring the message destination, you can configure one or more filters to determine which message types are written to that destination. A filter sorts messages based on the event type (log class) and the severity level.

The log class indicates the subsystem that generated the message. The severity indicates the lowest level message to display. You get messages of that class and below, with Emergency being the lowest and Debug the highest. If you set severity to **error**, you will receive Emergency, Alert, Critical and Error events. The following severity levels are recognized by AA-SBC:

Severity level
emerg (Emergency)
alert (Alert)
crit (Critical)
error (Error)
warning (Warning)
notice (Notice)
info (Information)
debug (Debug)

For a complete description of the event message types and severity levels, see *Net-Net OS-E – Using the NNOS-E Management Tools*.

services

Purpose

Opens the **services** object from where you configure event log settings and enable virus scanning. In addition, you define locations and schedule tasks for AA-SBC.

Syntax

```
config services
```

Properties

None

event-log

Purpose

Enables and disables global event-log administration. This control manages syslog, file, local and external database, and CLI event log storage or display.

Syntax

```
config services event-log
```

Properties

Property name	Description
admin {enabled | disabled}	Sets the global event-logging administrative state on this AA-SBC device. If **disabled**, all syslog, file system, and local-database configurations are ignored. **Example: set admin enabled** The default setting is **enabled.**

syslog

Purpose

Enables and disables a remote syslog server, specified by the syslog server IP address, and sets the filters to define which events AA-SBC sends.

Syntax

```
config services event-log syslog ipaddress
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables AA-SBC event logging to the remote syslog server. **Example: set admin enabled** The default setting is **enabled.**
filter *logClass severity*	Specifies the event message filter log class and severity level for messages forwarded to the syslog server. Repeat the command to specify multiple event filters. See Using filters with event log messages for complete information. **Example: set filter snmp warning** There are no default settings.
facility {user | local0...local7}	Sets the user-defined syslog facility (**user** or **local0** to **local7**) to which AA-SBC logs system events. Syslog facilities help isolate the origin of messages written to the syslog server. **Example: `set facility local5`** The default setting is **user**.

file

Purpose

Specifies AA-SBC configuration settings for the named event log file. Specify the name using up 64 alphanumeric characters with no blank spaces. Optionally, you can specify directory file paths using the forward slash (/) character. Additionally, you set the filters to define which events AA-SBC sends with this object.

Syntax

```
config services event-log file name
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables the system event log file. **Example: set admin enabled** The default setting is **enabled.**
`filter` *`logClass severity`*	Specifies the event filter type and severity level for messages written to the event log. Repeat the command specify multiple event filters. See Using filters with event log messages for complete information. **Example: set filter snmp warning** There is no default setting.
`size` *`megabytes`*	Set the maximum allowable size, in megabytes, of the event log file. **Example: set size 20** Enter a value between 1 and 100. The default setting is **10** MB.
`count` *`fileNumber`*	Specifies the maximum number of event log files to create when the event log file reaches the maximum size in megabytes. When the maximum count is reached, the first file in the rotation is cleared for rewriting and the count is resumed. **Example: set count 3** Enter a value between 1 and 20. The default setting is **5** event log files.

local-database

Purpose

Specifies the configuration settings for storing events in the AA-SBC local database and sets the filters to define which events AA-SBC sends.

Syntax

```
config services event-log local-database
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables the system local database. When **disabled**, the system does not write event log messages to the local database. However, you can still view any messages previously in the local database using the **show event-log** command. **Example: set admin enabled** The default setting is **enabled**.
filter *logClass severity*	Specifies the event filter type and severity level for messages written to the local database. Repeat the command to specify multiple event filters. See Using filters with event log messages for complete information. **Example: set filter snmp warning** There is no default setting.
history *days*	Sets the maximum number of days to store events in the local database. When the maximum number of days is reached, the local database is cleared and is restarted at the first day. **Example: set history 50** Enter a value between 1 and 10,000. The default setting is **100** days.

external-database

Purpose

Specifies the external (remote) database that serves as a target for event messages. Define this database with the database object. Additionally, you set the filters to define which events AA-SBC sends with this object.

For more information on the services/database object, refer to the *Net-Net OS-E – System Administration Guide*.

Syntax

```
config services event-log external-database name
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables the external database configuration. When **disabled**, the system does not write event log messages to the database. **Example: set admin enabled** The default setting is **enabled**.
filter *logClass severity*	Specifies the event filter type and severity level for messages written to the external database. Repeat the command to specify multiple event filters. See Using filters with event log messages for complete information. **Example: set filter snmp warning** There is no default setting.
history *days*	Sets the maximum number of days to store events in the external database. When the maximum number of days is reached, the database is cleared and is restarted at the first day. **Example: set history 50** Enter a value between 1 and 10,000. The default setting is **100** days.

cli

Purpose

Globally enables or disables writing of events to the CLI and sets the filters to define which events AA-SBC sends. Use the log-target action to enable or disable the feature for the current CLI.

Syntax

```
config services event-log cli
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables writing event messages to the CLI. **Example: set admin enabled** The default setting is **enabled**.
filter *logClass severity*	Specifies the event filter type and severity level for messages written to the CLI. Repeat the command to specify multiple event filters. See Using filters with event log messages for complete information. **Example: set filter snmp warning** There is no default setting.

smtp

Purpose

Enables and disables mailing of events to a designated SMTP server and sets the filters to define which events AA-SBC sends. AA-SBC then collects the events into an email and sends them to the SMTP server once every minute. Enter the host name or IP address of the SMTP server to open this object.

Syntax

```
config services event-log smtp host
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables the SMTP server event log archiving configuration. When **enabled**, the system emails session event logs to the specified address. When **disabled**, the system does not email the log files. **Example:** **`set admin enabled`** The default setting is **enabled**.
destination-mailbox *address*	Specifies the email address to which the system sends the session event logs. **Example: set destination-mailbox admin@companyABC.com** There is no default setting.
reply-mailbox *address*	Specifies the address that appears in the “From” field of the emailed event logs. If you do not specify this property, the “From” field is empty. **Example: set reply-mailbox events@companyABC.com** There is no default setting.
port *portNumber*	Specifies the port number over which the system should communicate with this SMTP server. **Example:** **`set port 100`** Enter a value from 1 to 65535; the default port is **25**.

Property name	Description
connection-keepalive *minutes*	Specifies the length of time, in minutes, that the system keeps the connection to the SMTP server open. This prevents opening and closing the connection with each event. **Example: `set connection-keepalive 15`** Enter a value between 5 and 60; the default setting is **5** minutes.
filter *logClass severity*	Specifies the event filter type and severity level for messages forwarded to the SMTP server. Repeat the command to specify multiple event filters. See Using filters with event log messages for complete information. **Example: set filter dns crit** There is no default setting.

tivoli

Purpose

Enables and disables sending of events to a designated Tivoli server and sets the filters to define which events AA-SBC sends. Enter the host name or IP address of the Tivoli server to open this object.

Syntax

```
config services event-log smtp host
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables the Tivoli server event log archiving configuration. When **enabled**, the system sends session event logs to the specified address. When **disabled**, the system does not send the log files. **Example: `set admin enabled`** The default setting is **enabled**.
protocol {tcp | udp}	Specifies the protocol the system uses to communicate with the Tivoli server. **Example: `set protocol udp`** The default protocol is **tcp**.
port *portNumber*	Specifies the port number over which the system communicates with this Tivoli server. **Example: `set port 7501`** Enter a value from 1 to 65535; the default port is 7500.
filter *logClass severity*	Specifies the event filter type and severity level for messages forwarded to the Tivoli server. Repeat the command to specify multiple event filters. See Using filters with event log messages for complete information. **Example: set filter dns crit** There is no default setting.

database

Purpose

Defines the external database. Configure AA-SBC to use this database as an external log target using the external-database object. Consult your database administrator for information regarding authentication on the remote database before configuring this object.

Syntax

```
config services database name
```

Properties

Property name	Description
driver *odbcDriver*	Specifies the name of the Open Database Connectivity (ODBC) driver associated with the database. **Example:** set driver psqlODBC There is no default setting.
username *name*	Specifies the user name needed for the system to access the database. This is the name the database expects to see when authenticating requests. **Example: set username nnos-e** There is no default setting.
secret-tag *string*	Specifies the secret tag (and password) needed for the system to access the database. This is the secret the database expects to see when authenticating requests. See Using filters with event log messages for information on the AA-SBC two-part password mechanism. **Example: set secret-tag 123** There is no default setting.
options *options*	Sets the options specified by the database. Use this to identify the location of the database. **Example: set options connection 192.168.100.100** There is no default setting.

instrument

Purpose

The instrumentation settings are for debugging and are intended for Technical Support use only.

data-locations

Purpose

Specifies the directory and path locations on AA-SBC where you would like to save certain types of information. This information includes:

- Accounting records prior to their being written to accounting targets.
- RTP media and mixed, for recording and playback of recorded calls.
- File transfer records.
- Log files.

Configuring these locations is optional; AA-SBC provides default directory path locations. If you choose not to configure locations, the default directory path for all file types is **/cxc_common/*** on the system hard-drive-1.

You can also configure multiple path locations. When you set a location, AA-SBC adds the location to the list of possible paths. AA-SBC uses these secondary locations when it reaches the fail-threshold (set with the storage-device object). You must use the **remove** command to delete an entry from the save/search list.

AA-SBC handles the location selection as follows:

- For files being written (call recordings, file transfer records, and log files), AA-SBC searches for an available location in the order in which the files were created. An available location is one that is mounted and not full.
- For files being read (call playback), AA-SBC searches all locations.

You can display the default directory file paths or the search order using the **show** command.The accounting process then reads those records in and services the various accounting targets. The file system acts as a large storage queue, providing two distinct benefits:

Saving accounting records

AA-SBC saves all accounting records to an internal files system before writing them to defined targets. Therefore, the accounting function has a more secure backup and more efficient record writing:

- Records can be saved until they are successfully written to the target. In addition, for a configurable amount of time they can be reapplied from the file system to the destination target if the target encounters problems, providing record recovery.
- Record writing is more efficient because it is not bound to call flow or dependent on target capabilities. The queueing mechanism of an external target could cause queue overflow and data loss. With an internal file system, files can be written to the target at a rate the target can handle.

The file system is made up of a root directory and subdirectories to hold the records. See the VSP accounting object for configurable options such as subdirectory size, purge criteria, and record retention periods.

Syntax

```
config services data-locations
```

Properties

Property name	Description
accounting-root-directory *filePath*	Sets the location where the system writes accounting records prior to their being sent to various configured accounting targets. For optimal record access, AA-SBC maintains an internal file structure of subdirectories within this specified root directory. Use the VSP accounting object to set the number of records stored in each subdirectory. **Example: set accounting-root-directory /acme_common/acct_records** The default path is **/cxc_common/accounting**.
rtp-recorded *filePath*	Sets the location in which the system saves call recordings. **Example: set rtp-recorded /acme_common/rtp_recorded** The default path is **/cxc_common/rtp_recorded**, with a backup location of **/cxc/recorded**.

Property name	Description
rtp-recorded-rotation {first-available | round-robin}	Sets the rotation scheme for writing recorded files to a directory. The system writes the files to the directories configured with the **rtp-recorded** property using one of the following schemes: • **first-available**—the system writes to the first directory listed with the **rtp-recorded** property that has enough space to hold the recording. The system will continue to write to that directory until the disk fills, and then it will move to the next directory in the list. • **round-robin**—the system rotates through all configured directories in a round-robin manner. This can allow an increase in the volume of simultaneous recorded calls by spreading the load across multiple disks. **Example: set rtp-recorded-rotation round-robin** The default is **first-available**.
rtp-mixed *filePath*	Sets the location in which the system writes for playback of recorded calls. This is where RTP files are “mixed” to create files that can then be played back. **Example: set rtp-mixed /acme_common/rtp_mixed** The default path is **/cxc_common/rtp_mixed**, with a backup location of **/cxc/mixed**.
file-transfer-recorded *filePath*	Sets the location in which the system saves file transfer records. **Example: set file-transfer-recorded /acme_common/ft_recorded** The default path is **/cxc_common/ft_recorded**, with a backup location of **/cxc/recorded**.

Property name	Description
log *filePath*	Sets the location in which the system saves log files. **Example: set log /acme/log** The default path is **/cxc_common/log**.
lnp-record-directory *filePath*	Provides a customer-specific application implementation and is not otherwise applicable. The default path is **/cxc_common/lnp**.

nfs

Purpose

Configures AA-SBC as a Network File System (NFS) client. NFS is Sun Microsystems' open protocol suite that allows computers running different operating systems to access shared files and share disk storage. It is the UNIX equivalent of Server Message Block (SMB). NFS allows users to import and export local files through an interface called the Virtual File System (VFS), which runs on top of TCP/IP.

When you enable an NFS entry, the AA-SBC acts as an NFS client, enabling remote mounted disk storage. Enter a name to open the nfs object. This specifies a mount point (a node in AA-SBC file directory). For example, if you name your **nfs** object alpha, then the external files are visible at /mnt/alpha/. You may want access to multiple servers, and/or multiple directories from a single server. In that case, each would have a separate NFS configuration object, with a unique mount point.

Syntax

```
config services data-locations nfs name
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disabled the mount point. **Example: set admin enabled** The default path is **enabled**.
server *hostname*	Specifies the IP address or host name of the NFS server. The system, as a NFS client, has access to the server file system. **Example: `set server 192.168.10.10`** There is no default server.
share *name*	Specifies the point in the file system that is being shared. When you configure an NFS server, you specify which directory is shared out (as well as read/write permissions and other properties). **Example: set share /home/staff/nfs** There is no default share.
version *versionNumber*	Specifies the version on NFS to use. AA-SBC supports versions 2 and 3. **Example: set version 2** The default version is version **3**.
protocol {udp | tcp}	Specifies the protocol to use when communicating with the server. **Example: set protocol tcp** The default protocol is **udp**.
timeout {default | custom *milliseconds*}	Specifies how long the system waits when trying to read from or write to the server. When the timeout value expires, the system cancels the action. If timeout is set to **default**, the driver determines the best value, depending on the version and/or protocol set. **Example: set timeout custom 5** The default is **default**. If you select custom, enter a number of seconds from 100 to 65535.

storage-device

Purpose

Sets the levels at which AA-SBC warns of approaching disk capacity and the frequency of those warnings. In addition, you set the level at which writes to the disk drive fail. If you have set backup file path locations (using the data-locations object), when a disk drive reaches the configured fail threshold setting, AA-SBC begins WRITE operations to the next available disk drive.

The **storage-device** object operates on all installed disk drives. If all disk drives have reached the configured free space threshold, media call recording, file transfers, and log files will no longer be written to AA-SBC disks.

Note: Currently, the AA-SBC devices support multiple disk drives.

Syntax

```
config services storage-device
```

Properties

Property name	Description
fail-threshold *megabytes*	Sets a threshold, in megabytes, at which the system no longer writes recorded calls or IM files to the disk drive. The system sends a warning message to the event log (and an SNMP trap) indicating that space on the internal disk drive has been exceeded. The system checks the fail threshold each time it receives a call. **Example: set fail-threshold 15000** Enter a value between 200 and 400,000. The default setting is **10000** MB.

tasks

Purpose

Opens the tasks object, from where you can configure and schedule archiving and maintenance tasks for a VSP.

Syntax

```
config services tasks
```

Properties

None

task

Purpose

Sets the action and the schedule for a task. A task can only contain one action. To schedule additional actions, create separate tasks. Each action uses its own set of arguments, described in the **arguments** property. (These are the same arguments you would supply when executing the function as an action instead of a task.)

Syntax

```
config services tasks task name
```

Properties

Property name	Description
schedule {disabled | period *hours* | time-of-day *time numberofDays* | days *time days* | once *time* [*date*]}	Sets the frequency with which the system executes the task. When entering the time (for **time-of-day** and **once**), you can enter the time in regular time format (for example, 3:00). The system displays the time in the format hh:mm:ss (for example, (03:00:00). Set either: • **disabled**—the task is not executed, but the configuration remains. • **period**—the task is executed with the specified frequency. Enter the interval, in hours from 1 to 288, between executions. • **time-of-day**—the task is executed at the specified time for the number of days specified, between 1 and 12. • **days**—the task is executed at the specified time on the days specified. • **once**—the task is executed once at the specified time. If you do not enter a date, the system uses the current day. **Example: set schedule time-of-day 1:00** The default setting is **disabled**.
action	Sets the action that the task performs. Each action has a dependent set of arguments. See descriptions below for complete action/argument details.
`arguments` *arguments*	Sets the arguments for the selected action. See descriptions below for complete action/argument details. Note that when you are entering more than one word or value for an argument you must enclose the string in quotation marks.

Property name	Description
action **archive**: arguments *vsp*Name	Saves stored sessions for the VSP. You must also enable archiving through the archiving object. See the archive action for more information. Note that if you have the **record-count** property of the archiving object set to any value other than 0, the archiving task will fail. **Example: set arguments vsp1** The default VSP archived is **default**.
action **call-failover** arguments flush	Flushes the call-failover database of any signaling and media-session records used to maintain call state between redundant AA-SBC devices. See the call-failover action for more information. **Example: set arguments flush** There is no default setting.
action **database** arguments {{delete | vacuum | vacuum-full | drop} *database* [*table*] | repair {translate | data-recovery} | initialize | snapshot {*integer* | force | automatic}}	Deletes or cleans database records (for databases you configured with the master services' database object). See the database action for more information. **Example: set arguments "snapshot log force"** There is no default setting.
action **database-maintenance**	Executes a multistep maintenance operation on entries found in the system database tables to optimize database access. See the database action for argument descriptions and more information. **Example: set arguments "save backup.xml"** Requires no arguments.
action **directory-clean**	Removes empty recorded media directories. You may have an empty directory, for example, if the system cleaned a directory as part of a scheduled maintenance operation. That action removes data but leaves the directories. This action takes no arguments.

Property name	Description
action **directory-reset** `arguments *vspName* [true | false]`	Resets the enterprise directory, causing the system to reread the directory and update the user information. Enter the name of the VSP that houses the directory. In addition, you can set a directory purge action of true or false: • **true**—clears out the contents of the database and then repopulates it. • **false**—updates the database but leaves users that are no longer in the directory itself in the database. **Example: set arguments “vsp1 false”** If you do not enter a VSP name, the system uses the VSP **default**. For the directory-reset action, the default purge action is **true**.
action **external-normalization** `arguments {replace-file *fileName* | replace-url *source* | flush}`	Manages the file used to maintain DNIS-to-ANI translation data. See the external-normalization action for more information. **Example: set arguments flush** There is no default setting.
action **external-presence** arguments {delete *url* | flush}	Clears all or a specified entry from the external presence cache. The external cache is the database running on the backup AA-SBC device in a cluster configuration. See the external-presence action for argument descriptions and more information. **Example: set arguments flush** There is no default setting.
action **external-session** arguments flush	Removes all entries from the external CSTA SIP session cache. See the external-session action for more information. **Example: set arguments flush** There is no default setting.

Property name	Description
action **file-based-word-lists-refresh**	Rereads any saved word-list or url-list **file** entry into memory. See the file-based-word-lists-refresh action for argument descriptions and more information. **Example: set arguments "delete 5085551212@abc.com"** This action takes no arguments.
action **file-transfer-delete-old** arguments *days*	Deletes all files brought on to the system, via a file transfer, that are older than the specified number of days. See the file-transfer-delete-old action for more information. **Example: set arguments 30** Enter a number of days between 1 and 1,000. The default number of days is 7.
action **install** arguments {file *source* [box | cluster | controlled] | url *source* [box | cluster | controlled] | nic [*model*] | nic-reinitialize | module | cancel}	Manages system software releases and network interface cards (NICs). See the install action for more information. **Example: set arguments "file release.tar.gz controlled"** There is no default setting.
action **load-balancing-failover** arguments flush	Deletes all recorded media files older than the specified number of days. See the media-delete-old action for more information. **Example: set arguments flush** There is no default setting.
action **location-database** arguments {merge [*filePath*] | replace [*filePath*] | save [*filePath*] | delete *aor* | flush}	Manages the location database across the cluster. See the location-database action for argument descriptions and more information. **Example: set arguments "save backup.xml"** For merge, replace, and save options, the default location is **/cxc/location.xml**.
action **loopback** arguments *{packet | packet-init} seconds to [from] [any | udp | tcp | tls]*	Establishes an outgoing SIP loopback call. See the loopback action for more information. **Example: set arguments packet 10 sip:5554443211@jane.cov.com** The default duration is 10 seconds.

Services objects

Property name	Description
action **media-delete-old** arguments *days*	Deletes all recorded media files older than the specified number of days. See the media-delete-old action for more information. **Example: set arguments 30** Enter a number of days between 1 and 1,000. The default number of days is 7.
action **orderly-restart** arguments {warm | cold | halt | cluster}	Causes a restart of the type specified after gracefully terminating any existing connections. See the orderly-restart action for argument descriptions and more information. **Example: set arguments cluster** The default type is **warm**.
action **presence** arguments {merge *fileName* | replace *fileName* | save *fileName* | delete *URL* | flush}	Manages the presence cache. See the presence action for argument descriptions and more information. **Example: set arguments “delete 5085551212@abc.com”** The default type is **warm**.
action **restart** arguments {warm | cold | halt | cluster}	Causes an immediate restart of the type specified. See the restart action for argument descriptions and more information. **Example: set arguments cluster** There is no default setting.
action **uri-alias** arguments {lookup *AOR* | reset | purge | seek *AOR*}	Causes an immediate restart of the type specified. See the uri-alias action for argument descriptions and more information. **Example: set arguments purge** There is no default setting.

network

Purpose

Sets system network parameters. By fine-tuning these settings, you gain greater control over network behavior. Generally the default TCP settings are adequate, however, so use caution before making changes.

Syntax

```
config services network
```

Properties

Property name	Description
tcp-keepalive-time *seconds*	Specifies the time, in seconds, that an established TCP connection can remain idle before the system sends a keepalive to the client. The idle time expiration initiates the keepalive process. **Example: set tcp-keepalive-time 1200** Enter a value between 30 and 14400; the default setting is **600** seconds.
tcp-keepalive-probes *integer*	Specifies the number of unanswered TCP keepalive probes that are allowed before the system disconnects an idle session. **Example: set tcp-keepalive-probes 3** Enter a value between 2 and 16; the default setting is **5** probes.
`tcp-keepalive-interval` *seconds*	Specifies the time, in seconds, that the system waits for a response from a keepalive probe before ending the next one. The system continues to send probes until it has sent the number specified in the **tcp-keepalive-probes** property. **Example: set `tcp-keepalive-interval 10`** Enter a value between 1 and 60; the default setting is **6** seconds.

Property name	Description
tcp-max-syn-backlog *integer*	Specifies the maximum number of queued (unacknowledged) connection requests allowed before the system begins dropping requests. This value is set to help prevent a TCP SYN flood attack. **Example: set tcp-max-syn-backlog 1536** Enter a value between 16 and 131072; the default setting is **1024** requests.
tcp-synack-retries *integer*	Specifies the number of times the system will retransmit a SYN-ACK in response to a SYN. If the number of retries is reached without a successful response, the system deletes the new connection from the table. This value helps minimize the effects of a SYN flooding attack. **Example: set tcp-synack-retires 4** Enter a value between 1 and 5; the default setting is **5** requests.
tcp-syncookies {enabled | disabled}	Enables or disables SYN cookie support in the kernel. When **enabled**, the kernel handles TCP SYN packets normally until the queue is full. Then, the kernel replies to a SYN with an intentionally modified TCP sequence number. A legitimate connection uses the number in the third packet of the three way handshake, allowing the system to verify and allow the connection, even though there is no corresponding entry in the SYN queue. An attacker would not respond with the sequence number and the connection is dropped. **Example: set tcp-syncookies disabled** The default setting is **enabled**.
tcp-fin-timeout *seconds*	Specifies the number of seconds the system waits for a final FIN packet before forcibly closing the socket. The system uses the FIN packet to disconnect a TCP connection, whether it's idle or not. **Example: set tcp-fin-timeout 120** Enter a value between 2 and 300; the default setting is **60** seconds.

monitors

Purpose

Opens the **monitors** object, through which you create monitoring configurations for tracking usage and TLS statistic threshold violations.

Syntax

```
config services monitors
```

Properties

None

monitor

Purpose

Sets threshold monitors for usage and TLS activity. When the threshold is exceeded, AA-SBC sends a message to the event log and creates an SNMP trap. You can set the interval at which AA-SBC polls the system and compares the current statistics against parameter thresholds.

Syntax

```
config services monitors monitor name
```

Properties

Property name	Description
interval *minutes*	Specifies the number of minutes the system waits between polls of the specified parameters. **Example: set interval 5** Enter a value between 1 and 60; the default setting is **10** minutes.
parameter {cpu-usage *percentage* | memory-usage *percentage* | kernel-memory-usage *percentage* | memory-failures *failures* | tls-connections *connections* | tls-failures *failures* | storage-devices *device percentage* | mos-failures *value* | syn-cookies *cookies* | dropped-media-packets *packets* | sip-parse-errors *errors*}	Sets the parameter to monitor and the threshold that, when exceeded, results in a message to the event log and an SNMP trap. Re-execute the command to add parameters. **Example: set parameter cpu-usage 90** **set parameter memory-usage 95** There is no default setting.

troubleshooting

Purpose

Sets the number of troubleshooting web service requests that can be handled by AA-SBC at one time. The object also sets an allowed wait time for pending requests.

Syntax

```
config services troubleshooting
```

Properties

Property name	Description
concurrent-requests *requests*	Specifies the number of concurrent web service troubleshooting requests the system attempts to service. If this threshold is reached, subsequent requests are queued for processing. They remain in the queue until: • they are processed because the queue dropped below the threshold. • they time out because they exceeded the maximum wait time assigned with the **concurrent-timeout** property. **Example: set concurrent-requests 4** Enter a value between 1 and 20; the default setting is **2** requests.
concurrent-timeout *milliseconds*	Specifies the maximum amount of time a troubleshooting request waits to be serviced before the system cancels the request. **Example: set concurrent-timeout 4000** Enter a value between 10 and 120,000; the default setting is **2000** milliseconds.

collect

Purpose

Configures the handling of data collection output files.

Syntax

```
config services collect
```

Properties

Property name	Description
directory	Specifies where the data collection output files will be stored. While the default directory is sufficient in most cases, if you are collecting the contents of a large database, this property allows you to specify a mount with more available disk space. **Example: set directory /cxc_common/collect_directory** The default setting is **/cxc_common/collect**.
max-old-files	Specify the maximum number of old files the AA-SBC saves before backups are deleted. The minimum valid value is 1 and the maximum valid value is 50. **Example: set max-old-files 25** The default setting is **5**.

default-collect-settings

Purpose

Enables or disables the default collection parameters. When one of these properties is set to disabled, the corresponding data is not collected.

Syntax

```
config services collect default-collect-settings
```

Properties

Property name	Description
config [enabled | disabled]	Enable or disable the collection of configuration data. **Example: set config disabled** The default setting is **enabled**.
certificates [enabled | disabled]	Enable or disable the collection of certificate data. **Example: set certificates disabled** The default setting is **enabled**.
status [enabled | disabled]	Enable or disable the collection of status data. **Example: set status disabled** The default setting is **enabled**.
crash-files [enabled | disabled]	Enable or disable the collection of crash-file data. **Example: set crash-files disabled** The default setting is **enabled**.
log-files [enabled | disabled]	Enable or disable the collection of log-files data. **Example: set log-files disabled** The default setting is **enabled**.
status-class	Specifies additional status classes to be collected. This property is a vector, so you can specify multiple entries. In addition, wildcards can be specified as well as the -v property to specify a verbose display in the status output file. **Example: set status-class location-bindings-rejected -v**

Property name	Description
database	Specifies the databases you want to collect. This property is a vector, so you can specify multiple entries. The following are valid databases you can collect: • log • spotlite • status • dos • directory • accounting Note: Use this property with caution as it is possible to specify the collection of enormous amounts of data. **Example: set database accounting**
directory	Specifies any additional directories you want collected. This property is a vector, so you can specify multiple entries. Note: Use this property with caution as it is possible to specify the collection of enormous amounts of data. **Example: set directory /cxc_common/data1/dir1**

collect-group

Purpose

Configures custom collection parameters as well as the default parameters.

Syntax

```
config services collect collect-group
```

Properties

Property name	Description
config [enabled | disabled]	Enable or disable the collection of config data for this collect-group. **Example: set config disabled** The default setting is **enabled**.
certificates [enabled | disabled]	Enable or disable the collection of certificate data for this collect-group. **Example: set certificates disabled** The default setting is **enabled**.
status [enabled | disabled]	Enable or disable the collection of status data for this collect-group. **Example: set status disabled** The default setting is **enabled**.
crash-files [enabled | disabled]	Enable or disable the collection of crash-file data for this collect-group. **Example: set crash-files disabled** The default setting is **enabled**.
log-files [enabled | disabled]	Enable or disable the collection of log-file data for this collect-group. **Example: set log-files disabled** The default setting is **enabled**.
status-class	Specifies additional status classes to be collected for this collect-group. This property is a vector, so you can specify multiple entries. In addition, wildcards can be specified as well as the -v property to specify a verbose display in the status output file. **Example: set status-class location-bindings-rejected -v**

Property name	Description
database	Specifies the databases you want to collect for this collect-group. This property is a vector, so you can specify multiple entries. The following are valid databases you can collect: • log • spotlite • status • dos • directory • accounting Note: Use this property with caution as it is possible to specify the collection of enormous amounts of data. **Example: set database accounting**
directory	Specifies any additional directories you want collected for this collect-group. This property is a vector, so you can specify multiple entries. Note: Use this property with caution as it is possible to specify the collection of enormous amounts of data. **Example: set directory /cxc_common/data1/dir1**

62. Session configuration objects

Session configuration description

The session configuration objects define the way in which AA-SBC handles SIP-based signaling and media traffic. The session configuration that is applied to an active call through AA-SBC depends on configuration of other aspects of the system.

There are several places in the configuration hierarchy through which you can access the session configuration objects. The path to these object defines in which cases AA-SBC uses that configuration. Locations for session configuration are defined in the following table.

Path	Defines...
vsp default-session-config	The session configuration settings to apply to those SIP calls for which there are no configured policies. See Chapter 18, “Default session configuration objects”, for more information.
vsp policies session-policies policy rule	The session configuration settings to apply to SIP calls for which a configured policy exists. See Chapter 48, “Policy objects”, for more information.
vsp dial-plan dial-prefix vsp dial-plan route vsp dial-plan source-route	The session configuration settings to apply to calls based on the dial prefix or domain suffix. See Chapter 21, “Dial plan objects”, for more information.
vsp calling-group route vsp calling-group source-route	The session configuration settings to apply to calling-group member calls based on the dial prefix or domain suffix. See Chapter 11, “Calling Group objects”, for more information.
vsp session-config-pool entry	A saved session configuration that can be referenced within one or more dial plans. See Chapter 63, “Session configuration pool objects”, for more information.

Session configuration object summary

The following table lists and briefly describes the session configuration objects. See the following chapter for other objects in the CLI hierarchy:

- Chapter 75, “Virtual system partition (VSP) objects”
- Chapter 18, “Default session configuration objects”
- Chapter 48, “Policy objects”
- Chapter 21, “Dial plan objects”
- Chapter 63, “Session configuration pool objects”

Object name	Description
`sip-settings`	Sets default SIP settings for calls without applicable policies.
`peer`	Sets the next-hop server based on session configuration.
`to-uri-specification`	Specifies what derives the content of the fields of the TO URI when AA-SBC transmits a message.
`from-uri-specification`	Specifies what derives the content of the fields of the FROM URI when AA-SBC transmits a message.
`request-uri-specification`	Specifies what derives the content of the fields of the REQUEST URI when AA-SBC transmits a message.
`location-normalization`	Normalizes the Request and/or To URI in a REQUEST message based on location cache data.
`location-lookup`	Customizes the manner in which AA-SBC executes a location lookup.
`location-events`	Sets the method by which the system implements bridged line appearance (BLA) for a phone.
`location-call-admission-control`	Customizes call admission control on a per-AOR basis.

Object name	Description
inbound-controls	Specifies what derives the content of the fields of the outgoing P-Asserted-Identity URI when AA-SBC transmits a message.
remote-party-id-specification	Configures the settings for header manipulation of the User portion of the Remote-Party-ID header of the SIP message.
contact-uri-settings-in-leg	Specifies where AA-SBC derives the content of the CONTACT header from when it forwards a message to a UAC.
contact-uri-settings-out-leg	Specifies where AA-SBC derives the content of the CONTACT header from when it forwards a message to a UAS.
media	Sets the default SIP media anchoring settings.
nat-traversal	Enables and disables symmetric RTP on AA-SBC.
recording-policy	Specifies whether to record a call and how to handle unsupported codecs.
media-scanner-settings	Configures media scanner settings, which monitor the signal strength and duration of received audio, to be divided into intervals.
transcoding-policy	Configures the AA-SBC transcoding policy.
periodic-announcement	Specifies a file and frequency of play to insert into a call.
media-verify-config	Enables media verification for RTP sessions and sets SIP session termination.
rtp	Enables or disables verification of the RTP header.
codec	Sets the payload and packet size and rate calculations for a codec.
rtcp-header	Enables or disables verification of the RTCP header.

Object name	Description
call-monitoring	Provides third-party conferencing, allowing a third-party participant to be added to a call in progress.
in-encryption	Specifies encryption parameters for inbound calls.
out-encryption	Specifies encryption parameters to outbound endpoints.
media-type	Sets the media types that are allowed and prohibited during the session.
bodypart-type	Sets the body types that are allowed and/or prohibited during the session.
dns-client-settings	Creates per-session configuration to apply to DNS clients.
in-codec-preferences	Sets a preference for CODECs, influencing the AA-SBC selection on the inbound leg of a call.
out-codec-preferences	Sets a preference for CODECs, influencing the AA-SBC selection on the outbound leg of a call.
in-media-normalization	Changes the media descriptor string for inbound calls.
out-media-normalization	Changes the media descriptor string for outbound calls.
in-hold-translation	Configures the SDP hold attributes that are sent to an endpoint that initiated a call.
out-hold-translation	Configures the SDP hold attributes that are sent to an endpoint that initially received a call.
in-dtmf-translation	Controls the method used for forwarding DTMF tones in inbound calls.
out-dtmf-translation	Controls the method used for forwarding DTMF tones in outbound calls.
sdp-regeneration	Sets parameters to “regenerate” the SDP.

Object name	Description
sip-directive	Sets default action for calls without applicable policies.
inbound-request-uri-specification	Specifies whether AA-SBC modifies the content of the host, port, and/or transport fields of the REQUEST URI.
contact-uri-settings-3xx-response	Specifies where AA-SBC derives the content of the CONTACT header in 3xx (response/redirect) messages that it receives.
emergency-settings	Allows an override for admission and emission control for matching calls.
calling-group-settings	Sets calling group membership and references an established group for settings.
presence	Enables and configures presence services.
registration	Renames the originating host and call user name for AA-SBC location service database.
authentication	Sets the default authentication mode to use on SIP sessions.
authorization	Sets the type of least cost routing (LCR) lookup that AA-SBC performs for matching sessions.
accounting	Sets the default accounting target and configuration path to apply to SIP sessions
log-alert	Enables/disables and configures session alert logging and writing of log details.
refer-settings	Enables or disables call parking compatibility settings for the Sylantro SIP for Business initiative.
group-settings	Creates a group that can later be referenced from within a condition-list.
instant-messaging	Enables IM archiving and configures text replacement and alerts.

Object name	Description
`instant-messaging-content`	Configures pointers to the word lists and URL lists used in instant message content scans.
`file-transfer`	Enables file transfer recording and virus scanning on transferred files.
`forking-settings`	Sets the ringing pattern for VoIP phones.
`header-settings`	Sets fields to remove and/or remove and replace in the SIP header.
`altered-header`	Modifies or creates whole header values in calls matching this session configuration.
`reg-ex-header`	Provides more granular modification capabilities to create or modify a header.
`header-normalization`	Modifies the User portion of the specified header.
`trusted-interface-settings`	Provides an interface to allow non-LCS devices to interact with LCS clients.
`session-control-settings`	Specifies whether AA-SBC should process policy on only the first or on all messages in a session.
`playback-call-settings`	Enables playback of the last recorded SIP call in a specific To/ From pair.
`csta-settings`	Provides CSTA-to-CAP communications for enterprises using a BroadWorks PBX and Microsoft LCS.
`sip-session-timers-settings`	Sets the values of SIP session timers.
`routing-settings`	Configures a geolocation to match against configured IP interfaces. Configures the call access mechanism in AA-SBC.
`pre-call-authorization`	Configures AA-SBC as a third-party call controller and specifies the WAV files that it should play.
`uui-header`	Modifies or creates the UUI header.

Object name	Description
handle-response	Specifies the action AA-SBC should take when it receives a specific response code from a call matching this session-config.
handle-publish	Sets whether events are sent to a third-party server via .
3GPP	Configures 3rd Generation Partnership Project (3GPP) systems.
Rx	Configures communication between AA-SBC and the Camiant Policy Server (3GPP Rx).
response-translation-settings	Maps status codes and phrases.
response-translation-settings	Adds a custom data field to the accounting call detail record (CDR).
codec-specific-parameters	Adds parameters to a specified CODEC for use in the a=fmtp line of SDP.
inbound-header-settings	Configures fields to remove and/or replace header settings in the SIP headers for inbound traffic.
header-normalization	Alters the user portion of a specified header.
altered-body	Alters the body of any SIP message for a matching session.
altered-header	Modifies or creates header values in calls matching this session configuration.

sip-settings

Purpose

Configures the SIP settings that AA-SBC applies to the SIP call session. If there are no configured policies or rules to enforce on the SIP call, then AA-SBC applies SIP settings from the default session configuration. If the call does match policy rules, AA-SBC applies the SIP settings defined in the session-config object.

Syntax

```
config vsp default-session-config sip-settings
config vsp policies session-policies policy name rule name
   session-config sip-settings
config vsp dial-plan dial-prefix entryName session-config sip-settings
config vsp dial-plan route name session-config sip-settings
config vsp dial-plan source-route name session-config sip-settings
config vsp session-config-pool entry name sip-settings
```

Properties

Property name	Description
mode {auto-determine | proxy}	Sets the SIP operating mode to use with this server Select either: • **auto-determine**—the system determines the mode. Usually, the system is a back-to-back user agent (B2BUA) that sends INVITE traffic to the call destination. (The system B2BUA appears as the SIP call destination, but regenerates the call to the destination SIP server.) In some cases, however, the system may act as a proxy instead. • **proxy**—the system is the SIP proxy that provides SIP registration, location, policy, and other services that determine the outcome of the SIP call. **Example: set mode proxy** The default setting is **auto-determine**.
transport {any | UDP | TCP | TLS}	Sets the default protocol over which the SIP call session is forwarded to the destination SIP server. Enter either UDP, TCP, TLS, or any. **Example: set transport TLS** The default setting is **any**.
port {auto-determine | override *port*}	Specifies the destination port on the system for SIP traffic. Select either: • **auto-determine**— the system sets the SIP port. • **override**—set the port number manually. Enter a port number or leave blank to accept the default setting of 5060. **Example: set port override 1212** The default setting is **auto-determine**. If you select **override**, enter a port number between 1 and 65535. The default port number for override is 5060.

Property name	Description
route-hdr {none | RouteRequest | RouteRequestResponse}	Sets if and where to insert a Record-Route header in relevant SIP requests for outgoing system interfaces. This allows the system to remain in the SIP signaling path. (Route headers are used by SIP proxies only, so this behavior affects only proxied traffic.) Select one of the following settings: • **none**—the system does not add Record-Route headers. • **RouteRequest**—the system inserts a Record-Route header in REQUESTs, allowing it to remain in the signaling path. • **RouteRequestResponse**—the system both inserts a Record-Route header in SIP REQUESTs and performs a Record-Route header fix in SIP RESPONSEs. This is needed when the SIP request and corresponding response are forwarded via different interfaces on the AA-SBC device. **Example: set route-hdr RouteRequestResponse** The default setting is **none**.
route-hdr-use-fqdn {enabled | disabled}	Specifies where the system derives the host portion of the Record-Route header from. When **enabled**, the system uses a domain name in the Record-Route header. That name is either: • configured with the **route-hdr-uri-host** property. • determined from a reverse lookup on the local address that the system used to transmit the SIP message. **Example: set route-hdr-use-fqdn disabled** The default setting is **enabled**.

Property name	Description
route-hdr-uri-host *string*	Specifies the string to insert in the host portion of the Record-Route header, identifying the AA-SBC domain name. This property is for use in conjunction with the **route-hdr-use-fqdn** property. If you set this, be sure that the string you enter is also set as the domain name or one of the domain aliases configured in the vsp static-stack-settings object. **Example: set route-hdr-uri-host cov.com** There is no default setting.
route-hdr-add-register-msg {enabled | disabled}	Specifies whether to add Record-Route headers to REGISTER messages. When **enabled**, the system adds Record-Route headers to REGISTER messages. Enable this for compatibility with Microsoft OCS 2007. When **disabled**, the system does not add the headers to Record-Route headers to REGISTER messages, in compliance with the suggestions of RFC 3261, SIP: Session Initiation Protocol. **Example: set route-hdr-add-register-msg enabled** The default setting is **disabled**.
route-hdr-preprocess-strip {enabled | disabled}	Controls whether or not the system strips the MAddr parameter off route headers prior to processing them. This property is only for use in cases where the system receives traffic from a non-RFC3261-compliant (strict-router) SIP proxy. Do not change the value unless instructed to do so by Technical Support. **Example: set route-hdr-preprocess-strip disabled** The default setting is **disabled**.

Property name	Description
lcs-compatibility {disabled | enabled [*value*]}	Enables or disables LCS compatibility. The specific bit setting required is dependent on the desired feature, and can only be determined for your system by Technical Support. For example, if you enable the **t120-anchor** property of the file-transfer object, you must set this bit to 0x010032e3. Do not enable this feature unless explicitly instructed to do so. **Example: set lcs-compatibility enabled 0x010032e3** The default setting is **disabled**.
in-server *serverType*	Specifies the originating server in a Sametime-to-LCS interaction. This allows the system to determine the transformations it must perform to facilitate communications. For the originating server, set the server version or function. The type that you select is dependent on the server type that you are configuring. Use this property in conjunction with the **out-server** property. For example, in an LCS-to-Sametime setup, the **in-server** type would be lcs-200*x* and the **out-server** would be sametime-31*x*. **Example: set in-server lcs-2005** The default setting is **unknown**.
out-server *serverType*	Specifies the receiving server in a Sametime-to-LCS interaction. This allows the system to determine the transformations it must perform to facilitate communications. For the receiving server, set the server version or function. The type that you select is dependent on the server type that you are configuring. Use this property in conjunction with the **in-server** property. For example, in an LCS-to-Sametime setup, the **in-server** type would be lcs-200*x* and the **out-server** would be sametime-31*x*. **Example: set out-server sametime-31** The default setting is **unknown**.

Property name	Description
utilize-contact {enabled | disabled}	Determines what the system should do with messages it sends out. When **enabled**, the system sets the address in the contact header to the AA-SBC local IP address. If **disabled**, the system does not form the From header from the Contact header. **Example: set utilize-contact disabled** The default setting is **enabled**.
add-contact-nat {enabled | disabled}	When **enabled**, the system appends the string "nat=true" to the contact header. If a phone is behind a firewall, and the system does not change the contact header to its own contact information, then it appends the string to the contact header. **Example: set add-contact-nat enabled** The default setting is **disabled**.
compress-signaling {enabled | disabled}	When **enabled**, the system applies gzip compression to all SIP messages between the endpoints. This feature is only applicable between two AA-SBC devices. **Example: set compress-signaling enabled** The default setting is **disabled**.
preserve-call-id {enabled | disabled}	Specifies whether the system generates a new call ID when forwarding an INVITE. Normally the system is a B2BUA, and therefore this property is **disabled**. As such, when it receives a call it generates a new ID for the outbound leg. If this property is **enabled**, the system uses the same call ID in the outbound INVITE that it received in the inbound INVITE. **Example: set preserve-call-id enabled** The default setting is **disabled**.

Property name	Description
preserve-cseq {enabled | disabled}	Specifies whether the system generates a new CSeq value when forwarding an INVITE. Normally the system is a B2BUA, and therefore this property is **disabled**. As such, when it receives a call it generates a new CSeq for the outbound leg. If this property is **enabled**, the system uses the same CSeq value in the outbound INVITE that it received in the inbound INVITE. **Example: set preserve-cseq enabled** The default setting is **disabled**.
proxy-generate-100-trying *msgType*	Specifies on which type of SIP messages the system should generate and send a “100 trying” message to the sender. The system passes the “100 trying” when it receives it for undeclared message types. By default, the system sends the “100 trying” as it was received (if the system is in proxy mode). Use this mode if the message contains information that must be passed from the SIP application server to the user agent. For example, additional headers in the message may contain version information necessary for proper operation of a softphone. When selected for a message type, the system immediately sends the initiator a “100 trying” to indicate that it is processing the message and will forward it as soon as possible. Sending an immediate response can prevent longer SIP messages from timing out. **Example: set proxy-generate-100-trying register+subscribe** There are no message types selected by default.

Property name	Description
handle-3xx-locally {enabled | disabled}	Specifies whether the system forwards responses to 3*xx* messages back to the UAC or resends it. If **disabled**, when the system receives a 3xx response for an INVITE (e.g., 300 Multiple Choices or 301 Moved Permanently), it forwards the response back to the UAC. When this option is **enabled**, the system does not forward the response back. Instead, it resends the INVITE message to the address specified in the contact header of the 3*xx* response. **Example: set handle-3xx-locally enabled** The default setting is **disabled**.
handle-3xx-locally-server-arbitration {enabled | disabled}	Specifies whether the system uses or ignores the dial-plan arbiter settings when forwarding a 302 Redirect message. The 302 contains the contact addresses to which the system redirects the message. If this property is **enabled**, AA-SBC applies the arbiter selected for the original INVITE to the contact addresses to determine the redirect destination. If **disabled**, the system uses the server qvalue (preference) to select the forwarding address. **Example: set handle-3xx-locally-server-arbitration enabled** The default setting is **disabled**.
handle-3xx-locally-lookup-original-invite {enabled | disabled}	Specifies whether, in the event of a 3xx response from a redirect server, the system modifies the original INVITE it receives before doing a dial-plan lookup. If **disabled**, the system does a lookup on the original INVITE. When this option is **enabled**, the system modifies the INVITE before initiating a lookup by replacing the Request URI in the INVITE with the Contact URI found in the 3xx response. You must enable this response if you are doing a source-route dial-plan lookup. If you do not, there will be no source on which to base the lookup. **Example: set handle-3xx-locally-lookup-original-invite enabled** The default setting is **disabled**.

Property name	Description
session-timeout *seconds*	Specifies how many seconds the system retains a session if the session did not establish successfully. For example, when authentication is required, case of authentication, the system sends a 401/407 response to the UAC to resend the request with authentication information. The session was not successful but cannot yet be terminated. This timer determines the length of time the system waits for a response. **Example: set session-timeout 45** Enter a value from 1 to 1,000,000; the default is **300** seconds.
session-duration-max *seconds*	Specifies how many seconds the system maintains a session after the session has been successfully established. This property puts a timer on the session and forces a close when the timer expires. If set to 0 (the default), the session remains open until it is complete. **Example: set session-duration-max 18000** Enter a value from 1 to 1,000,000; the default is **0** seconds (disabled).
session-provisional-timeout *seconds*	Specifies the number of seconds that the system allows the user agent server to ring before it times out the call. If forking-settings are configured, the system tries the next endstation when the timer expires, If not, the system terminates the call. If set to 0, the default, session termination due to unanswered ringing is determined by the user agent client. **Example: set session-provisional-timeout 30** Enter a value from 1 to 1,000,000; the default is **0** seconds (disabled).

Property name	Description
session-authentication-timeout *seconds*	Specifies the number of seconds that an INVITE session can stay in the authentication state before timing out. Because a first INVITE typically does not contain authentication information, it results in a 401 (Unauthorized) response. This timer sets the allowable time between that response and the second INVITE (which contains authentication information). If set to 0, the default, the timer is disabled and the sip-settings **session-timeout** property determines when the session times out. **Example: set session-authentication-timeout 30** Enter a value from 1 to 1,000,000; the default is **0** seconds (disabled).
outbound-local-ip *ipAddress*	Specifies which IP address to use to reach a destination when the AA-SBC tables contain multiple addresses to that destination. Use this in conjunction with the **outbound-local-port** property to specify an address and port combination. **Example: set outbound-local-ip 10.10.10.1** There is no default setting.
max-retransmissions *attempts*	Specifies a the maximum number of times the system attempts to retransmit SIP messages at the transaction level. By setting this value through the session config, you can apply different retransmission values to different message types. For example, you might want a higher number of retransmissions for a REGISTER message, and a lower number (faster response) for an INVITE. This value does not apply to OPTIONS messages. Use the settings **max-options-retransmissions** property to control OPTIONS retransmissions. **Example: set max-retransmissions 15** Enter a value from 1 to 32; the default setting is 1 attempt.

Property name	Description
outbound-local-port *port*	Specifies which port to use to reach a destination when the AA-SBC tables contain multiple ports to that destination. Use this in conjunction with the **outbound-local-ip** property to specify an address and port combination. **Example: set outbound-local-port 3435** There is no default setting.
message-session-timeout *seconds*	Specifies the number of seconds that the system keeps alive a session that was created by a MESSAGE request after the first transaction is complete. **Example: set message-session-timeout 2400** The default setting is **1800** seconds.
udp-source-port {auto-determine | override *port*}	Specifies the destination port on the system for UDP SIP traffic. Select either: • **auto-determine**— the system sets the UDP port. • **override**—set the port number manually. Enter a port number or leave blank to accept the default setting of 5060. **Example: set port override 1212** The default setting is **auto-determine**. If you select **override**, enter a port number between 1 and 65535. The default port number for override is 5060.
add-subject-header {none | header *string* {request-uri | to-uri | from-uri}}	Adds the specified string as a Subject header when a matched dial string is present in the user portion of the REQUEST, TO, or FROM URI. The system adds the header to all requests destined for devices that have successfully registered except for REGISTER, PRACK, ACK, BYE, and CANCEL. **Example: set add-subject-header header internalDevice=true to-uri** The default setting is **none**.

Property name	Description
strip-route-header {enabled | disabled}	Specifies whether the system should strip out the route header and ignore the information it contains on both inbound and outbound traffic. The route header is a field of the SIP header which is not commonly used, but when it is, the system gives it highest priority when forwarding calls. By default (when **disabled**), the system uses the information in the route header to forward packets. Set this property to **enabled** to ensure that manual configuration of SIP routing takes precedence. **Example: set strip-route-header enabled** The default setting is **disabled**.
strip-via-headers {enabled | disabled} *integer*	Specifies whether to remove all but the specified number of Via headers when forwarding a request. When **enabled**, the system strips all but the specified number of Via headers from the request as the request passes through the box. For example, if set to 1, the system strips all Via headers except the bottom (furthest). The system always inserts itself as the top Via header, regardless of the setting. It then restores the Via headers on the response. When **disabled**, the default, the system retains all Via headers. **Example: set strip-via-headers enabled 3** The default setting is **disabled**. If **enabled**, the default number of headers kept is 0 (strip all).
sticky-via {0 | 1 | 2}	Specifies whether to replace the port number in a VIA header. This property is only for use in cases where a unique port is assigned at connection and that port should be used in the VIA header. When this property is set to 0, the system leaves the port at the value that was received. When set to 1, the system replaces the value in the top VIA header. When set to 2, the system replaces the value in the next-to-last VIA header. Only use this property if instructed to do so by Technical Support. **Example: set sticky-via 1** The default setting is **0**.

Property name	Description
ignore-provisional-tag {enabled | disabled}	Specifies whether the system updates its internal SIP session using the TO header tag present in the provisional response sent by an intermediate server. If **enabled**, the system ignores the provisional response and updates the SIP session with the TO header tag found in the SIP final response. When **disabled**, the system updates the SIP session with the TO header from the provisional response. **Example: set ignore-provisional-tag enabled** The default setting is **enabled**.
forward-provisional-ack {enabled | disabled}	Specifies which points along the path respond to a provisional ACK. When **enabled**, the mid-stations along the path do not respond to the PRACK message. Only the endpoints respond (end-to-end). When set to **disabled**, the default, the each hop along the path responds. **Example: set forward-provisional-ack enabled** The default setting is **disabled**.
terminate-transaction-on-bye {enabled | disabled}	Specifies system behavior when it receives a BYE message. If **enabled**, the AA-SBC device terminates any outstanding transactions on that leg when it receives a BYE message. The system applies the termination to a particular call leg for a particular session. When **disabled**, the system does not terminate outstanding transactions when it receives a BYE. Instead, it only terminates them when either it receives the final response belonging to the transaction or the transaction times out. **Example: set terminate-transaction-on-bye enabled** The default setting is **enabled**.
to-header-follows-contact-header {enabled | disabled}	Specifies that the system should move the URI and header parameters from the Contact header to the To header. This property is used for a specific phone switching application only. **Example: set to-header-follows-contact-header enabled** The default setting is **disabled**.

Session configuration objects

Property name	Description
inleg-tos {preserve | overwrite [*value*]}	Determines the TOS value setting for the in-leg of the session. The TOS value determines the quality of service that the call receives. If set to **preserve**, the system uses the TOS value in the first received message of the session (normally a REGISTER or INVITE). If set to **overwrite**, the system marks the TOS field of all packets it sends out on the inleg with the value you specify. Enter a number that represents the 8-bit Differentiated Services (DS) field of the IP packet in decimal format, such as 26 for 00011010 or104 for 01101000. This value can be of use to upstream devices. **Example: set inleg-tos overwrite 22** The default setting is **preserve**. If you select **overwrite**, the default value is 0.
outleg-tos {preserve | overwrite [*value*]}	Determines the TOS value setting for the out-leg of the session. The TOS value determines the quality of service that the call receives. If set to **preserve**, the system uses the TOS value in the first received message of the session (normally a REGISTER or INVITE). If set to **overwrite**, the system marks the TOS field of all packets it sends out on the outleg with the value you specify. Enter a number that represents 8-bit Differentiated Services (DS) field of the IP packet in decimal format, such as 26 for 00011010 or 104 for 01101000. This value can be of use to upstream devices. **Example: set inleg-tos overwrite 22** The default setting is **preserve**. If you select **overwrite**, the default value is 0.
generate-final-response {enabled | disabled}	Sets whether the system generates a final response when it cannot forward an OPTIONS request or does not receive a response. When **enabled**, the system sends a 408 Request Timeout back to a UAC if it does not receive a response to an OPTIONS message from the UAS. When **disabled**, it does not generate a 408. **Example: set generate-final-response disabled** The default setting is **enabled**.

Property name	Description
b2bua-generate-100-trying {enabled | disabled}	Sets whether the system sends a 100 Trying message back to the caller before forwarding an INVITE. When **enabled**, the message is sent. **Example: set b2bua-generate-100-trying disabled** The default setting is **enabled**.
auto-accept-reinvite-with-no-sdp-on-in-leg {enabled | disabled}	Specifies whether the system responds with a 200 OK or forwards a REINVITE. When a call is first established, the system receives an INVITE on the in-leg and forward it out the out-leg. Once the call has been established, either side may send a REINVITE. If this property is **enabled** and the call is received on the in-leg, the system responds with a 200 OK. If **disabled**, the system forwards the REINVITE to the out-leg. **Example: set auto-accept-reinvite-with-no-sdp-on-in-leg enabled** The default setting is **disabled**.
auto-accept-reinvite-with-no-sdp-on-out-leg {enabled | disabled}	Specifies whether the system responds with a 200 OK or forwards a REINVITE. When a call is first established, the system receives an INVITE on the in-leg and forward it out the out-leg. Once the call has been established, either side may send a REINVITE. If this property is **enabled** and the call is received on the out-leg, the system responds with a 200 OK. If **disabled**, the system forwards the REINVITE to the in-leg. **Example: set auto-accept-reinvite-with-no-sdp-on-out-leg enabled** The default setting is **disabled**.

Property name	Description
strip-authint-qop {enabled | disabled}	Determines whether the system modifies the Quality of Protection (QoP) parameter. The QoP parameter defines the type of authentication the server requires, either auth or auth-int. Auth verifies the sender using a shared secret; auth-int verifies both the sender and the integrity of the message (as defined in RFC 2617, *HTTP Authentication: Basic and Digest Access Authentication*). When **enabled**, if the QoP parameter of a 401 or 407 challenge response offers both auth and auth-int, the system removes the auth-int option and forwards the message with just auth required. If **disabled**, the QoP parameter remains unchanged. **Example: set inleg-tos overwrite 22** The default setting is **disabled**.
ignore-cancel-branch {enabled | disabled}	Specifies whether to use the branch value to cancel a transaction. By default (**disabled**) the system uses the branch value (a string that identifies the transaction) in the top VIA header. When **enabled**, the system uses the FROM and TO tags in the call ID to identify and cancel the transaction. Use this property only if a device sends the wrong branch value. **Example: set ignore-cancel-branch enabled** The default setting is **disabled**.
symmetric-signaling {enabled | disabled}	Sets whether the system uses the VIA or CONTACT header to send messages for a call. An incorrectly functioning NAT device may only partially change a CONTACT or VIA header, causing the system to be unaware that the sending endpoint is behind a NAT. When **enabled**, this property ensures that the system always sends responses and future requests for that call to the IP address and port number from which the request originated. Enable this property only if instructed to do so by Technical Support. **Example: set symmetric-signaling enabled** The default setting is **disabled**.

Property name	Description
sips-uri-scheme-setting {auto | secure | not-secure}	Controls the SIP scheme the system uses in the URI of outbound calls. For example, when one endpoint is using TLS and another UDP, the SIP scheme is inconsistent. The default setting, **auto**, allows the system to determine the scheme. To force the setting, select either **secure** (for sips) or **not-secure** (for sip). **Example: set sips-uri-scheme-setting secure** The default setting is **auto**.
redirect-preserve-session-config {enabled | disabled}	Specifies where the system derives the session configuration from when a WSDL-generated INVITE results in a 3xx response. When **enabled**, the system applies the WSDL request's session configuration to the redirect message (the session configuration within the WSDL request is maintained throughout the session). When **disabled**, the default, the system uses the session configuration associated with the original INVITE for the redirect message as well, rather than the WSDL-based session configuration. **Example: set redirect-preserve-session-config enabled** The default setting is **disabled**.
max-forwards *integer*	Specifies the value to insert into the Max-Forward header of the SIP message. This value to determine how many future hops the message is allowed, and is decremented by 1 at each hop. If a message arrives with a Max-Forwards value higher than the value set with this property, the system resets the value to this configuration setting. If the value that arrives is equal to or lower than this setting, the system decrements that value by one. **Example: set max-forwards 50** The default setting is **70**.

Property name	Description
enum-fail-response {ignore | reject [*resultCode*] [*resultString*]}	Specifies the customized response code and string that the system sends as a result of a failed ENUM lookup. When set to **ignore**, the system ignores the failed lookup and continues to try and transmit the message. (The system may be able to resolve the address via a configured dial plan, for example.) If it is not able to ultimately resolve the TO header, the system sends a 404 Not Found message back to the sender. When set to **reject**, the system rejects the incoming message immediately and sends the configured code and string to the sender. You may use this, for example, if the originating phone/ gateway requires a custom cleardown (SIP response) when ENUM lookup fails. **Example: set enum-fail-response reject 999 “ENUM failure”** The default setting is **ignore**.
dns-fail-response-code *integer*	Sets the response code that the system sends to an endpoint when a call is received, the dns-client-settings **routing-last-resort-dns** property is applied, and the lookup fails. **Example: set dns-fail-response-code 400** The default setting is **404**.
dns-fail-response-string *string*	Sets the response string that the system sends to an endpoint when a call is received, the dns-client-settings **routing-last-resort-dns** property is applied, and the lookup fails. **Example: set dns-fail-response-string “Bad Request”** The default setting is **Not Found**.
update-moc-via-csta {enabled | disabled}	Specifies whether to send user state information relevant to the Microsoft Office Communicator (MOC) client. If **enabled**, the system notifies the MOC of the change in call state via CSTA when a call is connected or terminated. This provides MOCs that are logged in with remote-call-control enabled with correct presence information (in-a-call or available). **Example: set update-moc-via-csta enabled** The default setting is **disabled**.

Property name	Description
supported-inleg *string*	Adds a new Supported header or overwrites the existing header on the inbound leg of the SIP message with the specified text string. Use this property, for example, to support Provisional Response Acknowledgement (RFC 3262) PRACK insertion. **Example: set supported-inleg 100rel,timer** There is no default setting.
supported-outleg *string*	Adds a new Supported header or overwrites the existing header on the outbound leg of the SIP message with the specified text string. Use this property, for example, to support Provisional Response Acknowledgement (RFC 3262) PRACK insertion. **Example: set supported-outleg "Supported: 100rel"** There is no default setting.
persistent-destination-address {true | false}	Specifies whether to send a message to a new remote address based on a new remote contact. When set to true, the system ignores the Contact URI and forwards messages to their original destination address. When set to false, the system uses the Contact URI in each response to determine the destination of the next message. **Example: set persistent-destination-address false** The default setting is **true**.
loop-detection-threshold {enabled | disabled}	Specifies how many times a message can be processed by the system before it is considered to be in a loop. If this threshold is reached, the system rejects the call and sends a 482 (Loop Detected) response code back to the caller. **Example: set loop-detection-threshold 4** Enter a value between 1 and 256; the default setting is **2** passes.

Property name	Description
sip-signaling-encryption {unspecified | prohibited | required}	Specifies whether a call coming in over a TLS connection is allowed or dropped. Select either: • **unspecified**—any transport protocol is allowed; the system continues processing regardless of the connection type. • **prohibited**—TLS is not allowed. If a call comes in over a TLS connection, the system drops the call. • **required**—any transport protocol is allowed; the system continues processing regardless of the connection type. **Example: set sip-signaling-encryption required** The default setting is **unspecified**.
share-transport-connection {remote-ip| remote-addr | no-local-port | all}	Specifies when AA-SBC should reuse existing TCP or TLS connections. The setting specifies the criteria for matching between the existing connection and the current transaction (request or response). AA-SBC reuses the connection if the following match(es) occur: • **remote-ip**—the remote IP address. • **remote-addr**—the remote IP address and phone number. • **no-local-port**—the local IP address, the remote IP address, and the remote port. • **all**—the remote port and IP address and the local port and IP address. If there is no match, the system opens a new connection. **Example: set share-transport-connection all** The default setting is **remote-addr**.

Property name	Description
propagate-RR-headers {enabled | disabled}	*Secondary property.* Modifies headers in a B2B configuration. When **enabled**, the system propagates Record Route and Via headers from received to transmitted SIP messages. When **disabled**, the system creates these headers for each transmitted message. Do not modify this setting unless instructed to do so by Technical Support. **Example: set propagate-RR-header enabled** The default setting is **disabled**.
ignore-via-port {enabled | disabled}	*Secondary property.* Specifies whether to ignore the port reported in the top-level Via header. Typically, when the system matches a request to an existing transaction (request/response), it checks to make sure that, among other fields, the port is the same in both. This property should be **enabled** if there is a device along the path that incorrectly changes the Via header port. By default, this property is **disabled** (port numbers are checked). **Example: set ignore-via-port enabled** The default setting is **disabled**.
ignore-route-header {enabled | disabled}	*Secondary property.* Specifies whether AA-SBC uses the Route header to forward messages. By default (**disabled**), the system does use the Route header. When **enabled**, AA-SBC message forwarding logic does not use the Route header and instead uses other options, as set in the session configuration (Request URI, Contact URI, DNS, etc.). **Example: set ignore-route-header enabled** The default setting is **disabled**.

Property name	Description
make-provisional-resp-reliable *responseCode*	*Secondary property.* Specifies a single provisional response code to which the system will add “Required: 100rel” and “RSeq: *nnn*” before forwarding. These headers are only added, however, if the initial INVITE indicates support for 100rel. Note that the **forward-provisional-ack** property should be disabled when using this so that the PRACK from the UAC wont be forwarded to the UAS. If you are using third-party call control, the use183forRingingWithSdp converts a 183 response to a 180. In this case, set this property to 180 instead of 183. **Example: set ignore-route-header enabled** The default setting is **0**.
allow-redirect {enabled | disabled}	When enabled, the AA-SBC is able to redirect incoming calls to other servers. **Example: set allow-redirect disabled** The default setting is **enabled**.
strip-received-from-top-via {enabled | disabled}	When enabled, the AA-SBC strips off the “Received” and “Rport” fields in the top header before it sends a 200OK response. **Example**: **set strip-received-from-top-via enabled** The default setting is **disabled**.
preserve-session-config-on-3xx [enabled | disabled]	Apply the same session configuration that was used for the initial INVITE when sending the INVITE for a 302. When disabled, the AA-SBC removes the dial plan and server session configurations from the merged configuration. **Example: set preserve-session-config-on-3xx enabled** The default setting is **disabled**.

peer

Purpose

Sets the next-hop destination server where AA-SBC forwards a SIP messages. If this object is not set, calculations merged from the dial plan, location cache, and other policy settings control how AA-SBC forwards a message. These settings override the dynamic calculations, and effect the routing calculation that determines the next-hop server.

Syntax

```
config vsp default-session-config peer
config vsp policies session-policies policy name rule name
   session-config peer
config vsp dial-plan dial-prefix entryName session-config peer
config vsp dial-plan route name session-config peer
config vsp dial-plan source-route name session-config peer
config vsp session-config-pool entry name peer
```

Properties

Property name	Description
peer {trunk | switch | exchange} *reference rate* peer {server | calling-group | server-member} *reference* peer uac *contactString* {true | false} peer automatic peer named *carrier endpoint*	Specifies to which server the system should forward the call. Enter the name of a previously configured server or group of the type specified. • **trunk**—enter a reference to a carrier\exchange\switch trunk-group and optionally, a rate plan. • **switch**—enter a reference to a carrier exchange switch and optionally, a rate plan. • **exchange**—enter a reference to a carrier exchange and optionally, a rate plan. • **server**—enter a reference to an enterprise server. • **calling-group**—enter a reference to a calling-group group. • **server-member**—enter a reference to a server-pool server-pool-admission-control. • **uac**—allows you to specify any device, using the contact string that may be found in the location binding. To use the contact string in the location binding, use the **show location-cache -v** command. If the contact address is marked as **true**, it is stored as secure, using a SIPS: indicator. • **automatic**—sets the system to automatically route the session. In this case, the system bases routing decisions on settings in other objects (e.g., location cache, dial plan, and other policy settings). • **named**—identifies the next-hop server by matching on enterprise server carrier and endpoint tags. If multiple matches occur, the system hunts through all matches. To identify the server, enter the carrier tag used for an enterprise server and the endpoint tag used for the server-pool server-pool-admission-control. If the tags configured here do not match configured tags, no server is set as the next-hop, and the system responds with a 404 (not found) message. *Continued*

Property name	Description
peer *continued*	**Example:** **`set peer server “vsp enterprise servers sip-gateway CompanyABC”`** **`set peer uac sip:cov2078548357@elmaple.com true`** The default setting is **automatic**.
hunt-timeout *seconds*	Specifies the number of seconds to wait before determining a server is unavailable and trying the next configured server. **Example: set hunt-timeout 25** The default setting is **30** seconds.

to-uri-specification

Purpose

Specifies what derives the content of the fields of the TO URI when AA-SBC transmits a message. For example, if the **user** property is set to **to-uri**, AA-SBC replaces the user field of the TO URI with data from the user field of the incoming TO URI. If set to **omit**, the user field is left blank. Or, you can enter any string that you want placed in the user field.

Altering URIs

In some cases, it is necessary to change a portion of the URI (outbound only) so that the next-hop server can accept the SIP message when it arrives. For example, a server may require a part of the URI to be in a specific format. AA-SBC allows you to modify the portions of the REQUEST, TO, and/or FROM URI so that the header matches any necessary requirements.

When you select to alter the URI, you can set AA-SBC to replace the specified field with one of the following. Properties have some or all of the same selection options. These options apply to the to-uri-specification, from-uri-specification, request-uri-specification, and inbound-controls objects. The following defines the common options, where the AA-SBC device:

- **request-uri**—derives values from the incoming REQUEST URI.
- **to-uri**—derives values from the incoming TO URI.
- **from-uri**—derives values from the incoming FROM URI.
- **omit**—leaves the field blank.
- **next-hop**—derives values from the IP address of the next-hop server (not applicable to the **port** field).
- **local**—derives values from the IP address of the AA-SBC device (not applicable to the **transport** field).
- **omit-phone-context**—does not replace the field, but removes the phone context portion, if applicable (**user** property only).
- **next-hop-domain**—derives values from the IP address of the next-hop server or phone. If a server, the next-hop is learned from the peer property of the server; if a phone, it is the IP address of the phone. (**host** property only)
- **next-hop-ip**—derives values from the next-hop IP address, which is specified in the **peer** property of the same object. (**host** property only)
- **local-ip**—derives values from the IP address from the interface the packet goes out on. (**host** property only)
- **directory** *directoryReference*—derives values from a user alias. When selected, AA-SBC looks for all aliases associated with the user listed in the To, Request, and From fields of the URI. AA-SBC then uses the alias associated with the referenced directory. (**host** property only)
- *string*—writes the specified string to the field. (not applicable to the **transport** field)
- **none** or **same-uri**—the URI is not modified.

For example, it is not uncommon for a carrier to change a user ANI (automatic number identification) to the modified number used by the DNIS (dialed number identification service). For AA-SBC to correctly forward the call, it must put the user ANI back into the From header. It can do this if you configure the **host** property to a referenced directory that can identify the user DNIS alias.

Syntax

```
config vsp default-session-config to-uri-specification
config vsp policies session-policies policy name rule name
   session-config to-uri-specification
config vsp dial-plan dial-prefix entryName session-config
   to-uri-specification
config vsp dial-plan route name session-config to-uri-specification
config vsp dial-plan source-route name session-config
   to-uri-specification
config vsp session-config-pool entry name to-uri-specification
```

Properties

Property name	Description
user {request-uri | to-uri | from-uri | omit | next-hop | omit | local | omit-phone-context | *string*}	Specifies how to derive the value of the user field of the TO URI. **Example: set user request-uri** The default setting is **to-uri**.
host {request-uri | to-uri | from-uri | omit | next-hop | next-hop-domain | local-ip | *string*}	Specifies how to derive the value of the host field of the TO URI. **Example: set host request-uri** The default setting is **to-uri**.
port {request-uri | to-uri | from-uri | omit | *string*}	Specifies how to derive the value of the port field of the TO URI. **Example: set port omit** The default setting is **to-uri**.
display {request-uri | to-uri | from-uri | omit | next-hop | *string*}	Specifies how to derive the value of the display field of the TO URI. **Example: set display to-uri** The default setting is **to-uri**.
transport {request-uri | to-uri | from-uri | omit | UDP | TCP | TLS | next-hop}	Specifies the value of the transport field of the FROM URI. In addition to using the value from other fields of the incoming URI, you can set the transport method to UDP, TCP, or TLS. You cannot enter a string for this property. **Example: set transport TLS** The default setting is **to-uri**.

Property name	Description
user-param {omit | keep}	Specifies whether the User parameter in the TO URI of the SIP header is maintained or removed when the system forwards a message. If set to **keep**, the message is forwarded with the parameter as it was received. If set to **omit**, the entire user=*param* is removed from the TO URI. **Example: set user-param keep** The default setting is **omit**.
user-truncate-non-digits {enabled | disabled}	Specifies whether to remove non-digits from the User portion of the TO URI in INVITE messages. When **enabled**, the system removes all non-digits. **Example: set user-truncate-non-digits enabled** The default setting is **disabled**.
uri-parameter *name value* [append-always | append-if-does-not-exist | overwrite-existing]	Appends the specified user parameter and value to the TO URI for matching calls. For example, the example below would result in a TO URI that looked similar to: <sip:spot@fun.com;BTG=trunk1> You can control how and when the new parameter is added using the append options. Select **append-always** to have the parameter added to all matching calls. Select **append-if-does-not-exist** to add the name and value only if it does not already exist in the URI, preventing the possibility of duplicate parameters. Select **overwrite-existing** to replace any existing parameter in the TO URI with the configured name and value, updating instead of appending to the parameter. **Example: set uri-parameter BTG trunk1** You can append multiple user parameters. There is no default setting.

Property name	**Description**
header-parameter *name=value*	Adds a parameter string to the SIP header (outside of the SIP URI). Use this string, for example, to identify the source of a call or group destinations. You can add as many header parameters as required. Use the format name=value. To add multiple parameters use the format name=value;name=value... **Example: set header-parameter OLI=70** There is no default setting.
add-oli-tag *integer*	Specifies the number of digits to copy from the User portion of the To header to create an Originating Line Information (OLI) tag. The OLI tag provides information on the class of service available for the call. A value of 0 disables adding the tag. **Example: set add-oli-tag 2** The default setting is **0**.
strip-digits *integer*	Specifies the number of digits to strip from the User portion of the To URI. Use this, for example, if the original INVITE contains extra digits that would be problematic to the downstream server. Digits are removed beginning at the string that immediately follows the sip: or sips: portion. **Example: set strip-digits 2** The default setting is **0**.

from-uri-specification

Purpose

Specifies what derives the content of the fields of the FROM URI when AA-SBC transmits a message. For example, if the **user** property is set to **from-uri**, AA-SBC replaces the user field of the FROM URI with data from the user field of the incoming FROM URI. If set to **omit**, the user field is left blank. Or, you can enter any string that you want placed in the user field. See Altering URIs for information on replacement options.

Syntax

```
config vsp default-session-config from-uri-specification
config vsp policies session-policies policy name rule name
   session-config from-uri-specification
config vsp dial-plan dial-prefix entryName session-config
   from-uri-specification
config vsp dial-plan route name session-config from-uri-specification
config vsp dial-plan source-route name session-config
   from-uri-specification
config vsp session-config-pool entry name from-uri-specification
```

Properties

Property name	Description
user {request-uri | to-uri | from-uri | omit | next-hop | omit | local | omit-phone-context | *string*}	Specifies how to derive the value of the user field of the FROM URI. **Example: set user request-uri** The default setting is **from-uri**.
host {request-uri | to-uri | from-uri | omit | next-hop | next-hop-domain | local-ip | *string*}	Specifies how to derive the value of the host field of the FROM URI. **Example: set host request-uri** The default setting is **from-uri**.
port {request-uri | to-uri | from-uri | omit | *string*}	Specifies how to derive the value of the port field of the FROM URI. **Example: set port omit** The default setting is **from-uri**.
display {request-uri | to-uri | from-uri | omit | next-hop | *string*}	Specifies how to derive the value of the display field of the FROM URI. **Example: set display to-uri** The default setting is **from-uri**.

Property name	Description
user-agent-aware-display-translation {enabled | disabled}	Specifies whether or not to apply the displayname-character-set-info mappings to matching calls. When **enabled**, the system checks the display name for accented characters. If found, it does a location cache lookup on the destination. If that is successful, it checks the user agent found in the location cache against the configured mappings and performs any necessary translations. **Example: set user-agent-aware-display-translation enabled** The default setting is **disabled**.
transport {request-uri | to-uri | from-uri | omit | UDP | TCP | TLS | next-hop}	Specifies the value of the transport field of the FROM URI. In addition to using the value from other fields of the incoming URI, you can set the transport method to UDP, TCP, or TLS. **Example: set transport TLS** The default setting is **from-uri**.
user-param {omit | keep}	Specifies whether the User parameter in the FROM URI of the SIP header is maintained or removed when the system forwards a message. If set to **keep**, the message is forwarded with the parameter as it was received. If set to **omit**, the entire user=*param* is removed from the FROM URI. **Example: set user-param keep** The default setting is **omit**.
user-truncate-non-digits {enabled | disabled}	Specifies whether to remove non-digits from the User portion of the FROM URI in INVITE messages. When **enabled**, the system removes all non-digits. **Example: set user-truncate-non-digits enabled** The default setting is **disabled**.

Property name	Description
uri-parameter *name value* [append-always | append-if-does-not-exist | overwrite-existing]	Appends the specified user parameter and value to the FROM URI for matching calls. For example, the example below would result in a FROM URI that looked similar to: <sip:spot@fun.com;BTG=trunk1> You can control how and when the new parameter is added using the append options. Select **append-always** to have the parameter added to all matching calls. Select **append-if-does-not-exist** to add the name and value only if it does not already exist in the URI, preventing the possibility of duplicate parameters. Select **overwrite-existing** to replace any existing parameter in the FROM URI with the configured name and value, updating instead of appending to the parameter. **Example: set uri-parameter BTG trunk1** You can append multiple user parameters. There is no default setting.
header-parameter *name=value*	Adds a parameter string to the SIP header (outside of the SIP URI). Use this string, for example, to identify the source of a call or group destinations. You can add as many header parameters as required. Use the format name=value. To add multiple parameters use the format name=value;name=value... **Example: set header-parameter OLI=70** There is no default setting.
add-oli-tag *integer*	Specifies the number of digits to copy from the User portion of the From header to create an Originating Line Information (OLI) tag. The OLI tag provides information on the class of service available for the call. A value of 0 disables adding the tag. **Example: set add-oli-tag 2** The default setting is **0**.

Property name	Description
copy-charge-uri-user {enabled | disabled}	Specifies whether to copy the User portion of the Charge URI to the User portion of the From URI. When enabled, the system changes the content of the User portion, for example, for billing purposes. **Example: set copy-charge-uri-user enabled** The default setting is **disabled**.
strip-digits *integer*	Specifies the number of digits to strip from the User portion of the From URI. Use this, for example, if the original INVITE contains extra digits that would be problematic to the downstream server. Digits are removed beginning at the string that immediately follows the sip: or sips: portion. **Example: set strip-digits 2** The default setting is **0**.
prepend-digits *string*	Specifies a string to prepend to the User portion of the From URI. The string can be comprised of up to 128 ASCII characters. **Example: set prepend-digits 011** Enter a maximum of 128 characters. There is no default setting.

request-uri-specification

Purpose

Specifies what derives the content of the fields of the REQUEST URI when AA-SBC transmits a message. For example, if the **user** property is set to **to-uri**, AA-SBC replaces the user field of the REQUEST URI with data from the user field of the incoming TO URI. If set to **omit**, the user field is left blank. Or, you can enter any string that you want placed in the user field. See Altering URIs for information on replacement options.

Syntax

```
config vsp default-session-config request-uri-specification
config vsp policies session-policies policy name rule name
   session-config request-uri-specification
config vsp dial-plan dial-prefix entryName session-config
   request-uri-specification
config vsp dial-plan route name session-config
   request-uri-specification
config vsp dial-plan source-route name session-config
   request-uri-specification
config vsp session-config-pool entry name request-uri-specification
```

Properties

Property name	Description
user {request-uri | to-uri | from-uri | omit | next-hop | omit | local | omit-phone-context | *string*}	Specifies how to derive the value of the User field of the REQUEST URI. **Example: set user from-uri** The default setting is **request-uri**.
host {request-uri | to-uri | from-uri | omit | next-hop | next-hop-domain | local-ip | *string*}	Specifies how to derive the value of the Host field of the REQUEST URI. **Example: set host request-uri** The default setting is **request-uri**.
port {request-uri | to-uri | from-uri | omit | next-hop | *string*}	Specifies how to derive the value of the Port field of the REQUEST URI. **Example: set port omit** The default setting is **request-uri**.

Property name	Description
transport {request-uri | to-uri | from-uri | omit | UDP | TCP | TLS | next-hop}	Specifies the value of the Transport field of the REQUEST URI. In addition to using the value from other fields of the incoming URI, you can set the host transport method to UDP, TCP, TLS, or the method used by the next-hop server. **Example: set transport next-hop** The default setting is **request-uri**.
user-param {omit | keep}	Specifies whether the User parameter in the REQUEST URI of the SIP header is maintained or removed when the system forwards a message. If set to **keep**, the message is forwarded with the parameter as it was received. If set to **omit**, the entire user=*param* is removed from the REQUEST URI. **Example: set user-param keep** The default setting is **omit**.
user-truncate-non-digits {enabled | disabled}	Specifies whether to remove non-digits from the User portion of the REQUEST URI in INVITE messages. When **enabled**, the system removes all non-digits. **Example: set user-truncate-non-digits enabled** The default setting is **disabled**.

Property name	Description
uri-parameter *name value* [append-always | append-if-does-not-exist | overwrite-existing]	Appends the specified user parameter and value to the REQUEST URI for matching calls. For example, the example below would result in a REQUEST URI that looked similar to: <sip:spot@fun.com;BTG=trunk1> You can control how and when the new parameter is added using the append options. Select **append-always** to have the parameter added to all matching calls. Select **append-if-does-not-exist** to add the name and value only if it does not already exist in the URI, preventing the possibility of duplicate parameters. Select **overwrite-existing** to replace any existing parameter in the REQUEST URI with the configured name and value, updating instead of appending to the parameter. **Example: set uri-parameter BTG trunk1** You can append multiple user parameters. There is no default setting.

Property name	Description
apply-to-routing {true | false}	Specifies whether the system uses the REQUEST URI changes that result from the settings for routing and normalization or for normalization only. When set to **false**, the default, the system only changes the REQUEST URI for the next-hop SIP server (normalization). When **true**, the system uses the information in the modified REQUEST URI to forward the message (routing), bypassing the functions of the dial plan and/or location cache. Set this property to **true** only when the system is operating in proxy mode and relies on the modified REQUEST URI to correctly forward the message. For example, set this property to **true** in a CSTA application to cause the system to use the new REQUEST URI to resolve routing. **Example: set apply-to-routing true** The default setting is **false**.
use-location-cache-contact-uri {false | true}	Specifies whether the system should use the location cache information or the **request-uri-specification** configuration to modify the contact field of the REQUEST message. This property is only applicable when the system uses the location service to forward the request. When **true**, the system copies the phone contact URI (as found in the location cache) to the request URI. When **false**, the system uses the settings of this object to determine the content of the REQUEST message. **Example: set use-location-cache-contact-uri false** The default setting is **true**.

location-normalization

Purpose

Normalizes the Request and/or To URI in a REQUEST message. AA-SBC replaces the URI portion of the specified headers with the AOR data found in the location cache. With the properties of this object you can update the To and Request headers with the full AOR or just the host portion from the AOR in the location cache. This configuration provides a means of normalizing requests destined for an AOR (e.g., phone).

Syntax

```
config vsp default-session-config location-normalization
config vsp policies session-policies policy name rule name
   session-config location-normalization
config vsp dial-plan dial-prefix entryName session-config
   location-normalization
config vsp dial-plan route name session-config location-normalization
config vsp dial-plan source-route name session-config
   location-normalization
config vsp session-config-pool entry name location-normalization
```

Properties

Property name	Description
request-uri {none | AOR | AOR-host-only}	Specifies the portion of the AOR to use in normalizing the Request URI. You can replace the AOR portion of the Request URI with the entire AOR from the location cache or the domain name only (AOR-host-only). By default, no replacement occurs. **Example: set request-uri AOR** The default setting is **none**.
to-uri {none | AOR | AOR-host-only}	Specifies the portion of the AOR to use in normalizing the To URI. You can replace the AOR portion of the Request URI with the entire AOR from the location cache or the domain name only (AOR-host-only). By default, no replacement occurs. **Example: set to-uri AOR** The default setting is **none**.

location-lookup

Purpose

Customizes the manner in which AA-SBC executes a location lookup. Within this object you can specify which headers AA-SBC uses to perform the lookup, and in what order the fields should be searched on. Use this, for example, to send INVITEs to an AOR registered on AA-SBC but without using the AOR of this device in the Request URI.

Syntax

```
config vsp default-session-config location-lookup
config vsp policies session-policies policy name rule name
   session-config location-lookup
config vsp dial-plan dial-prefix entryName session-config
   location-lookup
config vsp dial-plan route name session-config location-lookup
config vsp dial-plan source-route name session-config location-lookup
config vsp session-config-pool entry name location-lookup
```

Properties

Property name	Description
sequence {request-uri | to-uri | from-uri | contact | trunk-port}	Selects the header field(s) used to do the lookup in the location cache. Enter as many headers as needed by re-executing the command. **Example: set sequence request-uri** There is no default setting.
match-uri-params *string*	Specifies which URI parameters should be used in the location lookup comparison. By default, the system does a base comparison of the received contact using the following URI information: • scheme (i.e., sip, sips, tel) • user • host • port (5060 if not otherwise specified) • transport (UDP if not otherwise specified) • maddr Enter a string for AA-SBC to match against. **Example: set match-uri-params state** There is no default setting.

location-events

Purpose

Sets the method by which the system implements bridged line appearance (BLA) for a phone. BLA is a feature that allows a phone to display line appearance (status) on more than one phone. A single telephone number (or SIP URL) can be monitored by more than one user agent, and when a call is made to the number, all subscribed user agents can respond. The status of the call is then displayed on all phones mapped to the number. This object sets the mechanism used to do that mapping, either the default SIP mechanism, or the method put forth in the IETF draft, *Implementing Bridged Line Appearances Using Session Initiation Protocol (SIP)*.

Syntax

```
config vsp default-session-config location-events
config vsp policies session-policies policy name rule name
   session-config location-events
config vsp dial-plan dial-prefix entryName session-config
   location-events
config vsp dial-plan route name session-config location-events
config vsp dial-plan source-route name session-config location-events
config vsp session-config-pool entry name location-events
```

Properties

Property name	Description
bridged-line-appearance {default | draft-anil-sipping-bla}	Sets the type of bridged line appearance the phone uses. Select default for most phones. Select draft-anil-sipping-bla for the following configurations: • when using a Sylantro phone. • when using sticky trunk ports, where the registration-plan route **alter-contact** property is set to trunk-port-per-aor, -endpoint, or -binding. **Example: set bridged-line-appearance draft-anil-sipping-bla** The default setting is **default**.

location-call-admission-control

Purpose

Customizes call admission control on a per-AOR basis. These parameters can also be configured in the location service settings object. From that object, the values are applied across the VSP. The settings override the VSP-wide settings and are applied when an AOR registers. The parameters remain in effect until the next registration period. Use the show location-cac status provider to view call admission control settings and counters for an AOR.

Syntax

```
config vsp default-session-config location-call-admission-control
config vsp policies session-policies policy name rule name
   session-config location-call-admission-control
```

```
config vsp dial-plan dial-prefix entryName session-config
   location-call-admission-control
config vsp dial-plan route name session-config
   location-call-admission-control
config vsp dial-plan source-route name session-config
   location-call-admission-control
config vsp session-config-pool entry name
   location-call-admission-control
```

Properties

Property name	Description
max-number-of-concurrent calls *integer*	Specifies the maximum number of active incoming and outgoing calls allowed for this AOR at one time. When this value is reached, the connection does not accept calls until the value drops below the threshold. **Example: set max-number-of-concurrent-calls 1500** Enter a value between 0 and 1,000,000; the default is **1000** calls. A value of 0 causes the system to decline all calls and registrations.
max-calls-in-setup *integer*	Sets the maximum number of simultaneous inbound and outbound call legs in setup stage that are allowed for this AOR. A call leg in setup is much more compute-intensive than established call legs, so this value is more restrictive than the concurrent call leg value. A value of 0 causes the system to decline all calls and registrations. **Example: set max-calls-in-setup 50** Enter a value between 0 and 10,000; the default is **30** call legs.
admission-control {enabled | disabled}	Specifies whether the system considers AOR limitations when forwarding a call *from* the AOR. The system tracks the number of concurrent (both incoming and outgoing) active calls for this AOR. If this property is **enabled**, the system does not forward calls from the AOR if the limit has been reached and instead sends a “603 Declined” message. If **disabled**, the system does forward calls from the AOR. (Set the call limit with the **max-number-of-concurrent-calls** property.) See Admission control for an AOR for additional information. **Example: set admission-control enabled** The default setting is **disabled**.

Property name	Description
max-bandwidth {unlimited | *kbps*}	*Secondary property.* Specifies the amount of bandwidth the system allocates to the AOR. When the system reaches the maximum bandwidth limit for a server, it rejects calls until bandwidth use drops below the maximum. **Example: set max-bandwidth 512** The default setting is **unlimited**.

Property name	Description
emission-control {enabled | disabled}	Specifies whether the system considers AOR limitations when forwarding a call *to* this AOR. The system tracks the number of concurrent (both incoming and outgoing) active calls for the AOR. If this property is **enabled**, the system does not forward calls to the AOR if the limit, set with the max-number-of-concurrent-calls property, has been reached. Instead, the system sends one of the following messages and drops the call: • If there is one outbound server/UAC/UAS, the system sends a “486 Busy” message, indicating that the route was resolved but that the AOR was unavailable. • If there are multiple outbound server/UAC/UASs and all have reached the maximum concurrent calls threshold, the system sends a “486 Busy” message. • If there are multiple outbound server/UAC/UASs and at least one has not reached the maximum concurrent calls threshold, the return code is determined by the final server that the system attempted to reach. This could be, for example, “486 busy” or a “504 server timeout” if the last server was unresponsive and the transaction timed out. If **disabled**, the system continues to forward calls to the AOR. See Admission control for an AOR for more information. **Example: set emission-control enabled** The default setting is **disabled**.

Property name	Description
call-rate-limiting {enabled [*callsPerInterval*] [*interval*] [*resultCode*] [*resultString*] | disabled}	*Secondary property.* Limits the number of calls sent to an AOR within a certain interval. Once this interval is reached, the system rejects any calls to this AOR until the rate decreases, returning a response code and message. This feature sets the acceptable arrival rate for incoming calls when used with **admission-control** and the acceptable set-up rate when used with **emission-control**. If **enabled**, set the number of calls allowed and the measurement interval (in seconds). You can also enter a result code from 400 to 699 and a text string to accompany call rejection if no available server is found. **Example: set call-rate-limiting enabled 50 1 480 "Temporarily unavailable"** The default value is **disabled**. If set to **enabled**, the default calls-per-interval is 60, the default interval is 1 second, and the default result is 486, Busy Here.

inbound-controls

Purpose

Customizes call admission control, on a per-AOR basis, for inbound calls only. This is in contrast to the admission control settings found in the location-call-admission-control object, which sets aggregate inbound and outbound limits. Use the **show location-cac** status provider to view call admission control settings and counters for an AOR.

Syntax

```
config vsp default-session-config location-call-admission-control
inbound-controls
config vsp policies session-policies policy name rule name
session-config location-call-admission-control inbound-controls
config vsp dial-plan dial-prefix entryName session-config
location-call-admission-control inbound-controls
config vsp dial-plan route name session-config
location-call-admission-control inbound-controls
config vsp dial-plan source-route name session-config
```

```
location-call-admission-control inbound-controls
config vsp session-config-pool entry name
location-call-admission-control inbound-controls
```

Properties

Property name	Description
max-number-of-concurrent calls *integer*	Specifies the maximum number of active incoming calls allowed for this AOR at one time. When this value is reached, the connection does not accept calls until the value drops below the threshold. **Example: set max-number-of-concurrent-calls 1500** Enter a value between 0 and 1,000,000; the default is **1000** calls. A value of 0 causes the system to decline all calls and registrations.
max-calls-in-setup *integer*	Sets the maximum number of simultaneous inbound call legs in setup stage that are allowed for this AOR. A call leg in setup is much more compute-intensive than established call legs, so this value is more restrictive than the concurrent call leg value. A value of 0 causes the system to decline all calls and registrations. **Example: set max-calls-in-setup 50** Enter a value between 0 and 10,000; the default is **30** call legs.

Property name	Description
max-bandwidth {unlimited | *kbps*}	*Secondary property.* Specifies the amount of bandwidth the system allocates to inbound calls for the AOR. When the system reaches the maximum bandwidth limit for an AOR, it rejects calls until bandwidth use drops below the maximum. **Example: set max-bandwidth 512** The default setting is **unlimited**.
call-rate-limiting {enabled [*callsPerInterval*] [*interval*] [*resultCode*] [*resultString*] | disabled}	*Secondary property.* Limits the number of calls sent to the AOR within a certain interval. Once this interval is reached, the system rejects any calls to this AOR, returning a response code and message, until the rate decreases. This feature sets the acceptable arrival rate for incoming calls. If **enabled**, set the number of calls allowed and the measurement interval (in seconds). You can also enter a result code from 400 to 699 and a text string to accompany call rejection if no available server is found. **Example: set call-rate-limiting enabled 50 1 480 “Temporarily unavailable”** The default value is **disabled**. If set to **enabled**, the default calls-per-interval is 60, the default interval is 1 second, and the default result is 486, Busy Here.

outbound-controls

Purpose

Customizes call admission control, on a per-AOR basis, for outbound calls only. This is in contrast to the admission control settings found in the location-call-admission-control object, which sets aggregate inbound and outbound limits. Use the **show location-cac** status provider to view call admission control settings and counters for an AOR.

Syntax

```
config vsp default-session-config location-call-admission-control
outbound-controls
config vsp policies session-policies policy name rule name
session-config location-call-admission-control outbound-controls
config vsp dial-plan dial-prefix entryName session-config
location-call-admission-control outbound-controls
config vsp dial-plan route name session-config
location-call-admission-control outbound-controls
config vsp dial-plan source-route name session-config
location-call-admission-control outbound-controls
config vsp session-config-pool entry name
location-call-admission-control outbound-controls
```

Properties

Property name	Description
max-number-of-concurrent calls *integer*	Specifies the maximum number of active outgoing calls allowed for this AOR at one time. When this value is reached, the connection does not accept calls until the value drops below the threshold. **Example: set max-number-of-concurrent-calls 1500** Enter a value between 0 and 1,000,000; the default is **1000** calls. A value of 0 causes the system to decline all calls and registrations.
max-calls-in-setup *integer*	Sets the maximum number of simultaneous outbound call legs in setup stage that are allowed for this AOR. A call leg in setup is much more compute-intensive than established call legs, so this value is more restrictive than the concurrent call leg value. A value of 0 causes the system to decline all calls and registrations. **Example: set max-calls-in-setup 50** Enter a value between 0 and 10,000; the default is **30** call legs.

Property name	Description
max-bandwidth {unlimited | *kbps*}	*Secondary property.* Specifies the amount of bandwidth the system allocates to outbound calls to the AOR. When the system reaches the maximum bandwidth limit for a server, it rejects calls until bandwidth use drops below the maximum. **Example: set max-bandwidth 512** The default setting is **unlimited**.
call-rate-limiting {enabled [*callsPerInterval*] [*interval*] [*resultCode*] [*resultString*] | disabled}	*Secondary property.* Limits the number of calls sent from the AOR within a certain interval. Once this interval is reached, the system rejects any calls from this AOR, returning a response code and message until the rate decreases. This feature sets the acceptable set-up rate for incoming calls. If **enabled**, set the number of calls allowed and the measurement interval (in seconds). You can also enter a result code from 400 to 699 and a text string to accompany call rejection if no available server is found. **Example: set call-rate-limiting enabled 50 1 480 “Temporarily unavailable”** The default value is **disabled**. If set to **enabled**, the default calls-per-interval is 60, the default interval is 1 second, and the default result is 486, Busy Here.

p-asserted-identity-uri-specification

Purpose

Specifies what derives the content of the fields of the outgoing P-Asserted-Identity URI when AA-SBC transmits a message. For example, if the **user** property is set to **to-uri**, the system replaces the user field of the P-Asserted-Identity URI with data from the user field of the incoming TO URI. If set to **omit**, the user field is left blank. Or, you can enter any string that you want placed in the user field. See Altering URIs for information on replacement options.

Syntax

```
config vsp default-session-config p-asserted-identity-
   uri-specification
config vsp policies session-policies policy name rule name
   session-config p-asserted-identity-uri-specification
config vsp dial-plan dial-prefix entryName session-config
   p-asserted-identity-uri-specification
config vsp dial-plan route name session-config
   p-asserted-identity-uri-specification
config vsp dial-plan source-route name session-config
   p-asserted-identity-uri-specification
config vsp session-config-pool entry name p-asserted-identity-
   uri-specification
```

Properties

Property name	Description
user {same-uri | request-uri | to-uri | from-uri | omit | next-hop | *string*}	Specifies how to derive the value of the user field of the P-Asserted-Identity URI. **Example: set user from-uri** The default setting is **same-uri**.
host {same-uri | request-uri | to-uri | from-uri | omit | next-hop | next-hop-domain | local-ip | *string*}	Specifies how to derive the value of the host field of the P-Asserted-Identity URI. **Example: set host request-uri** The default setting is **same-uri**.

Property name	Description
port {same-uri | request-uri | to-uri | from-uri | omit | *string*}	Specifies how to derive the value of the port field of the P-Asserted-Identity URI. **Example: set port omit** The default setting is **same-uri**.
display {same-uri | request-uri | to-uri | from-uri | omit | next-hop | *string*}	Specifies how to derive the value of the display field of the P-Asserted-Identity URI. **Example: set display to-uri** The default setting is **same-uri**.
transport {same-uri | request-uri | to-uri | from-uri | omit | UDP | TCP | TLS | next-hop}	Specifies the value of the transport field of the P-Asserted-Identity URI. In addition to using the value from other fields of the incoming URI, you can set the transport method to UDP, TCP, TLS, or the method used by the next-hop server. You cannot enter a string for this property. **Example: set transport next-hop** The default setting is **same-uri**.
user-param {omit | keep}	Specifies whether the User parameter in the P-Asserted-Identity URI of the SIP header is maintained or removed when the system forwards a message. If set to **keep**, the message is forwarded with the parameter as it was received. If set to **omit**, the entire user=*param* is removed from the P-Asserted-Identity URI. **Example: set user-param keep** The default setting is **omit**.
uri-parameter *name value*	Appends the specified user parameter and value to the P-Asserted-Identity URI for matching calls. The resulting parameter is added to the URI. For example, the example below would result in a P-Asserted-Identity URI that looked similar to: <sip:jane@fun.com;BTG=trunk1> **Example: set uri-parameter BTG trunk1** You can append multiple user parameters. There is no default setting.

Property name	Description
create {true | false}	Sets whether to create a P-Asserted-Identity header if one does not already exist. If **true**, the system uses the property settings in this object to determine the value of the fields in the header. If set to **false**, the system does not create the header. If a header does already exist, this field has no effect. (The fields of the existing header are still manipulated by the values of the object's properties.) **Example: set create true** The default setting is **false**.
use-original-from-header {true | false}	Specifies where the content of the P-Asserted-Identity header is derived from. (This property is only applicable if the **create** property is set to **true**.) When **use-original-from-header** is **true**, the system uses the From header as it existed, before any normalization occurred, as the p-asserted-identity header. If set to **false**, the system uses the From header as it exists when SIP call processing occurs (e.g., post normalization). **Example: set use-original-from-header true** The default setting is **false**.

remote-party-id-specification

Purpose

Configures the settings for header manipulation of the User and/or Host portion of the Remote-Party-ID header of the SIP message. Within this object you can modify the existing header or create a new header if one is not present. You can then modify the content by prepending or stripping off characters, for example, to ensure interoperability between carriers. Typically these headers are only part of INVITE messages.

Syntax

```
config vsp default-session-config remote-party-id-specification
config vsp policies session-policies policy name rule name
   session-config remote-party-id-specification
config vsp dial-plan dial-prefix entryName session-config
   remote-party-id-specification
config vsp dial-plan route name session-config
   remote-party-id-specification
config vsp dial-plan source-route name session-config
   remote-party-id-specification
config vsp session-config-pool entry name
   remote-party-id-specification
```

Properties

Property name	Description
user-source {remote-party-id-uri | from-uri | to-uri | request-uri | contact-uri}	Specifies the URI from which the system extracts the User portion and applies it to the Remote-Party-ID header—either From, To, Request, or Contact. If left at the default, **remote-party-id-uri**, there is no change to the USER portion. For example, if you select from-uri, the system extracts the User portion of the From URI and applies it to overwrite or create the User portion of the existing or newly created Remote-Party-ID header. **Example: set user-source from-uri** The default setting is **remote-party-id-uri**.
user-action {none | prepend *phonePrefix* | strip-off number | replace-with *newNumber*}	Sets the action to take once the system has derived the content of the Remote-Party-ID header. You can prepend, strip-off, or replace digits to the User portion. You can also make no modifications. (For example, you may want to create a header using the remote-party-id-specification object, but not want to make any modifications to the default header that is created.) **Example: set user-action prepend 1** The default setting is **none**.

Property name	**Description**
host-source {remote-party-id-uri | from-uri | to-uri | request-uri | contact-uri | text *string*}	Specifies where the system extracts the Host portion for the new or existing Remote-Party-ID header. Set the host field to be derived from the remote-party-id, To, From, Request, or Contact URIs of the INVITE, or you can enter any text string. **Example: set host-source to-uri** The default setting is **remote-party-id-uri**.
create {true | false}	Sets whether to create a Remote-Party-ID header if one does not already exist. If **true**, the system uses the property settings in this object to determine the value of the fields in the header. If set to **false**, the system does not create the header. If a header already exists, this field has no effect. (The fields of the existing header are still manipulated by the property values. **Example: set create true** The default setting is **false**.

contact-uri-settings-in-leg

Purpose

Specifies where AA-SBC derives the content of the CONTACT header from when it forwards a message to a UAC. For example, if the **user** property is set to **to-uri**, AA-SBC replaces the User field of the CONTACT header with data from the User field of the incoming TO header. The inbound leg of the session is the portion from AA-SBC to the call initiator (UAC).

Note that this modification does not apply to REGISTER requests. To make changes to the headers of a REGISTER request, use the properties of the registration-plan object.

Syntax

```
config vsp default-session-config contact-uri-settings-in-leg
config vsp policies session-policies policy name rule name
   session-config contact-uri-settings-in-leg
config vsp dial-plan dial-prefix entryName session-config
   contact-uri-settings-in-leg
config vsp dial-plan route name session-config
   contact-uri-settings-in-leg
config vsp dial-plan source-route name session-config
   contact-uri-settings-in-leg
config vsp session-config-pool entry name contact-uri-settings-in-leg
```

Properties

Property name	Description
user {request-uri | to-uri | from-uri | contact-uri | omit | *string*}	Specifies how to derive the value of the User field (the resource located at host) of the CONTACT header. • **request-uri**—uses the value from the incoming REQUEST URI. • **to-uri**—uses the value from the incoming TO URI. • **from-uri**—uses the value from the incoming FROM URI. • **next-hop**—uses the IP address of the next-hop server. • **omit**—leaves the field blank. • **string**—writes the specified string to the field. **Example: set user from-uri** The default setting is **contact-uri**.
host {cxc-address | public-address | original-address | next-hop-address | *string*}	Specifies how to derive the value of the Host field (the host providing SIP resource) of the CONTACT header. • **cxc-address**—uses the AA-SBC IP address as the host • **public-address**—uses the public address for a UAC behind a firewall or the UAC address if it is not behind a firewall. • **original-address**—the host field is not modified. • **next-hop-address**—uses the IP address of the next-hop server. However, this value is typically used only with the contact-uri-settings-3xx-response object. • **string**—writes the specified string to the field. **Example: set host original-address** The default setting is **cxc-address**.

Property name	Description
port {cxc-local-port | original-port | omit | *string*}	Specifies how to derive the value of the Port field (where the request is to be sent) of the CONTACT header. • **cxc-local-port**—uses the port number that the system transported the call over. • **original-port**—the port field is not modified. • **omit**—leaves the field blank. • **string**—writes the specified string to the field. **Example: set port original-port** The default setting is **cxc-local-port**.
transport {next-hop-transport | original-transport | omit | UDP | TCP | TLS}	Specifies the derivation of the transport type for the Transport field of the CONTACT header. • **next-hop-transport**—uses the method used by the next-hop server. • **original-transport**—the transport field is not modified. • **omit**—leaves the field blank. • **UDP, TCP, TLS**—sets the transport field to the selected protocol. **Example: set transport original-transport** The default setting is **next-hop-transport**.
add-maddr {enabled | disabled}	Specifies whether to include the MAddr parameter in the CONTACT header. If enabled, and the HOST field of the header is a FQDN, the system adds its own IP address as the MAddr parameter. If **disabled**, or if the HOST field is not a FQDN, the system replaces the HOST with its IP address. **Example: set add-maddr disabled** The default setting is **enabled**.

Property name	Description
use-incoming-contact {enabled | disabled}	Determines the basis for creating the CONTACT header in an outbound message. When **enabled**, the system first copies the content of the inbound header to build the outbound header. Using the inbound header content, the system then applies any further changes defined in this contact-uri-settings-in-leg object. **Example: set use-incoming-contact enabled** The default setting is **disabled**.
from-user-contact-uri {enabled | disabled}	This property is not applicable to the inbound leg of a connection. **Example: set from-user-contact-uri enabled** The default setting is **disabled**.
registration-plan-precedence {true | false}	Sets whether or not the system applies the CONTACT header modifications specified within this object. When set to **true**, the system does not apply changes to messages if a registration-plan is present. When set to **false**, all message types are changed. **Example: set registration-plan-precedence false** The default setting is **true**.

Property name	Description
add-other-params {enabled | disabled}	Specifies whether the system maintains additional parameters in incoming CONTACT headers. When **enabled**, the system allows any additional parameters that were received in the CONTACT header to remain in the new CONTACT header when it rewrites it as a result of matching this session config. Additional (or “other”) parameters are those found after the URI. For example, in the header “Contact: <sip:johnD@10.1.1.1:5060 udp>;team,” “team” is the other parameter and remains in the new CONTACT header. When **disabled**, the system removes additional parameters. **Example: set add-other-params enabled** The default setting is **disabled**.
always-include-contact-header {enabled | disabled}	Sets the system verify whether there is a contact header present in each message. When enabled, the system checks to ensure that there is a Contact header present. If there is not, it creates one. The content of the header is derived from the other properties in this object. When **disabled**, the system does not check for the presence of the Contact header. **Example: set always-include-contact-header enabled** The default setting is **disabled**.

contact-uri-settings-out-leg

Purpose

Specifies where AA-SBC derives the content of the CONTACT header from when it forwards a message to a UAS. For example, if the **user** property is set to **to-uri**, AA-SBC replaces the User field of the CONTACT header with data from the User field of the outgoing TO header. The outbound leg of the session is the portion from AA-SBC to the call responder (UAS).

Note that this modification does not apply to REGISTER requests. To make changes to the headers of a REGISTER request, use the properties of the registration-plan object. Also, if you have enabled enum-apply-request-result-to-contact in the dial-plan normalization object, you must set this object so that the Contact header is not overwritten on the outbound side. To do so, set **use-incoming-contact** to **enabled**, set host, port, and transport to use their original values, and set user to the Contact URI.

Syntax

```
config vsp default-session-config contact-uri-settings-out-leg
config vsp policies session-policies policy name rule name
   session-config contact-uri-settings-out-leg
config vsp dial-plan dial-prefix entryName session-config
   contact-uri-settings-out-leg
config vsp dial-plan route name session-config
   contact-uri-settings-out-leg
config vsp dial-plan source-route name session-config
   contact-uri-settings-out-leg
config vsp session-config-pool entry name contact-uri-settings-out-leg
```

Properties

Property name	Description
user {request-uri | to-uri | from-uri | contact-uri | omit | *string*}	Specifies how to derive the value of the User field (the resource located at host) of the CONTACT header. • **request-uri**—uses the value from the incoming Request URI. • **to-uri**—uses the value from the incoming To URI. • **from-uri**—uses the value from the incoming From URI. • **contact-uri**—uses the value from the incoming Contact URI. Select this value to preserve changes made to the Contact header through the dial-plan normalization object. • **omit**—leaves the field blank. • **string**—writes the specified string to the field. **Example: set user from-uri** The default setting is **contact-uri**.
host {cxc-address | public-address | original-address | next-hop-address | *string*}	Specifies how to derive the value of the Host field (the host providing SIP resource) of the CONTACT header. • **cxc-address**—uses the AA-SBC IP address as the host • **public-address**—uses the public address for a UAC behind a firewall or the UAC address if it is not behind a firewall. • **original-address**—the host field is not modified. Select this value to preserve changes made to the Contact header through the dial-plan normalization object. • **next-hop-address**—uses the IP address of the next-hop server. However, this value is typically used only for the contact-uri-settings-3xx-response object. • **string**—writes the specified string to the field. **Example: set host original-address** The default setting is **cxc-address**.

Property name	Description
port {cxc-local-port | original-port | omit | *string*}	Specifies how to derive the value of the Port field (where the request is to be sent) of the CONTACT header. • **cxc-local-port**—uses the port number that the system transported the call over. • **original-port**—the port field is not modified. Select this value to preserve changes made to the Contact header through the dial-plan normalization object. • **omit**—leaves the field blank. • **string**—writes the specified string to the field. **Example: set port original-port** The default setting is **cxc-local-port**.
transport {next-hop-transport | original-transport | omit | UDP | TCP | TLS}	Specifies the derivation of the transport type for the Transport field of the CONTACT header. • **next-hop-transport**—uses the method used by the next-hop server. • **original-transport**—the transport field is not modified. Select this value to preserve changes made to the Contact header through the dial-plan normalization object. • **omit**—leaves the field blank. • **UDP, TCP, TLS**—sets the transport field to the selected protocol. **Example: set transport original-transport** The default setting is **next-hop-transport**.
add-maddr {enabled | disabled}	Specifies whether to include the MAddr parameter in the CONTACT header in an outbound message. If **enabled**, and the HOST field of the header is a FQDN, the system adds its own IP address as the MAddr parameter. If **disabled**, or if the HOST field is not a FQDN, the system replaces the HOST with its IP address. **Example: set add-maddr disabled** The default setting is **enabled**.

Property name	Description
use-incoming-contact {enabled | disabled}	Determines the basis for creating the CONTACT header in an outbound message. When **enabled**, the system first copies the content of the inbound header to build the outbound header. Using the inbound header content, the system then applies any further changes defined in this contact-uri-settings-out-leg object. Set this property to **enabled** to preserve changes made to the Contact header through the dial-plan normalization object. **Example: set use-incoming-contact enabled** The default setting is **disabled**.
from-user-contact-uri {enabled | disabled}	Specifies whether the system uses the location cache to derive the CONTACT header when forwarding a message. When **disabled**, the default, the CONTACT URI is derived from the FROM header of the original message. When this property is **enabled**, the system does a location cache lookup on the received FROM URI. If the system finds an entry, it uses the server-side contact (found in the entry) as the CONTACT URI for the outbound message. **Example: set from-user-contact-uri enabled** The default setting is **disabled**.
registration-plan-precedence {true | false}	Sets whether or not the system applies the CONTACT header modifications specified within this object to REGISTER requests. When set to **true**, the system does not apply changes to REGISTER messages. When set to **false**, all message types are changed. **Example: set registration-plan-precedence false** The default setting is **true**.

Property name	Description
add-other-params {enabled | disabled}	Specifies whether the system maintains additional parameters in incoming CONTACT headers. When **enabled**, the system allows any additional parameters that were received in the CONTACT header to remain in the new CONTACT header when it rewrites it as a result of matching this session config. Additional (or “other”) parameters are those found after the URI. For example, in the header “Contact: <sip:johnD@10.1.1.1:5060 udp>;team,” “team” is the other parameter and remains in the new CONTACT header. When **disabled**, the system removes additional parameters. **Example: set add-other-params enabled** The default setting is **disabled**.
always-include-contact-header {enabled | disabled}	Sets the system verify whether there is a contact header present in each message. When enabled, the system checks to ensure that there is a Contact header present. If there is not, it creates one. The content of the header is derived from the other properties in this object. When **disabled**, the system does not check for the presence of the Contact header. **Example: set always-include-contact-header enabled** The default setting is **disabled**.

media

Purpose

Configures the SIP media anchoring settings to apply to this SIP call session.

Note: You must define and enable a pool of ports specifically for media services. Without these ports defined, AA-SBC cannot establish the anchoring necessary to provide media services. See the media-ports object for more information.

RTCP settings

Real-time Control Protocol (RTCP) is a companion protocol to RTP that gathers statistics on the performance and quality of the SIP call connection. When enabled, RTP monitors the quality of the SIP call and conveys information about the SIP call session. It is based on the periodic transmission of control packets to all participants in the session, and provides feedback on the quality of the data distribution.

RTCP statistics are used to dynamically adjust and optimize the call quality for current network conditions. You can configure AA-SBC to drop, pass, and generate RTCP packets and to record the statistical data in its database through the **media** object.

Media session maintenance

AA-SBC provides a session maintenance feature for use in cases of abnormal session terminations. Normally, SIP signaling causes the creation and termination of each media session. In the event of network or device failure, however, terminating SIP messages may not be received by the signaling system, or the signaling system may be unable to request that the resources be released by the media proxy. You can enable the **inactivity-timeout** property to recover resources for aborted media sessions.

When **inactivity-timeout is** enabled, the media proxy periodically checks for inactive media sessions. If a session timed out due to inactivity, a message is sent to the signaling AA-SBC device, which logs the event and sends the appropriate SIP signaling messages to notify each party of the call. The media proxy then releases the resources for the inactive session.

Transcoding media types

AA-SBC supports transcoding media types, which is the process of converting media from one CODEC into a different CODEC on output. This allows, in some cases, endpoints supporting different media types to communicate. You add CODECs using the **transcode-media-types** property, augmenting the list of CODECs contained in an INVITE SDP offer. Note that the order in which a CODEC appears in the offer/answer matters in the AA-SBC selection. Original media types appear first, followed by the media types added in this object. You can re-order the media types with the move command, or using the in-codec-preferences or out-codec-preferences objects.

The following example illustrates and explains the transcoding process.

Device	CODEC
Phone East	Supports A, B, Z
Phone West	Supports B, C, Z
NNOS-E	Configured with transcoding support for C
NNOS-E	Does not support Z

As a result, the following sequence occurs:

1. EAST's original INVITE/SDP offering contains CODECs A, B, Z.
2. AA-SBC augments the list with CODEC C, resulting in an offer of A, B, Z, C in the INVITE/SDP.
3. WEST's original OK/SDP response answers with CODECs B, C, Z.
4. AA-SBC modifies and forwards the response to EAST with CODECs B, Z.

With this configuration, AA-SBC:

AA-SBC action	Reason
Forwards Z packets in both directions unchanged.	Because CODEC Z is unsupported.
Transcodes C packets from WEST to B on their way to EAST.	Because B is the first supported CODEC in the response (on behalf of WEST) to EAST. That response is comprised of the WEST response minus CODECs that AA-SBC added. (Note that if there are no remaining CODECs after this process, AA-SBC adds back in the first CODEC common to EAST and the system.)
Forwards B packets from either direction, no transcoding necessary.	Because both sides understand B.

Use the **rtp-transcode-stats** and **rtp-transcode-summary** status providers to view active and summary statistics for RTP transcoding.

Syntax

```
config vsp default-session-config media
config vsp policies session-policies policy name rule name
   session-config media
config vsp dial-plan dial-prefix entryName session-config media
config vsp dial-plan route name session-config media
config vsp dial-plan source-route name session-config media
config vsp session-config-pool entry name media
```

Properties

Property name	Description
anchor {enabled | disabled | auto}	Enables or disables SIP media session anchoring on this SIP call session. Media anchoring forces the SIP media session to traverse the system. The **auto** setting enables conditional anchoring. In this case, the system uses its auto-anchoring algorithms to determine anchoring necessity based on a variety of criteria, including whether you have configured smart anchoring via the autonomous-ip object and whether the calling devices are behind a firewall. **Example: set anchor enabled** The default setting is **enabled**.
transcode-media-types *inputCodec*	Specifies the CODEC(s) that the system can use for transcoding media. Adding CODECs through this property augments the list contained in an INVITEs SDP offer. See Transcoding media types for a full description and example. You can enter any number of valid input CODEC types. To see a list of available input CODECs, type a question mark at the command line. **Example: set transcode-media-types g726-16** There is no default setting.

Property name	Description
auto-conference [enabled | disabled] *regExp* [in | out | both]	Allows a user to log into a conference call automatically by prepending the assigned username and passcode before the number to be dialed. To do this, the SIP session establishes the initial call, and then identified DTMF tone strings are used to join the conference, creating the conference codes. To configure, set: • **administrative state**—turns on or off the auto-conference feature. When **enabled**, the system strips the Request URI and plays the first portion of the regular expression as tones. • **regular expression**—identifies the digit strings in the user portion of the Request URI. All but the last item matched are played as tones after the call comes up. The system replaces the last match as the outgoing user portion in the Request URI. • **direction**—specifies which side of the call hears the tone string. For example, if the Request URI is 1234567#123#johnd@webex.com, the resulting digit strings played are 1234567# and 123#. The Request URI becomes johnd@webex.com. Use the **pre-** and **post-tone-delay** properties to allow announcements to play from the conference site before the digits are entered. Use the VSP dtmf-generation object to set parameters for the conference codes created. **Example: set auto-conference enabled (.*#)(.*#)(.*) out** The default setting is administratively **disabled**, with no regular expression specified, and the outbound side hearing the tones.

Property name	Description
pre-tone-delays *milliseconds*	Specifies the milliseconds of delay prior to playing the DTMF tone string that was determined from the **auto-conference** property. You can specify multiple delays to correspond to multiple matches of the regular expression. If the number of matches exceeds the number of delay entries, the last delay entry is repeated. For example, if you have three matches in the regular expression, and have configured delays of three and five milliseconds, the system delays 3 milliseconds, plays tone 1, delays 5 milliseconds, plays tone 2, delays 5 milliseconds, plays tone 3. **Example: set pre-tone-delays 3** There is no default tone delay.
post-tone-delays *milliseconds*	Specifies the milliseconds of delay following the playing of a DTMF tone string that was determined from the **auto-conference** property. You can specify multiple delays to correspond to multiple matches of the regular expression. If the number of matches exceeds the number of delay entries, the last delay entry is repeated. For example, if you have three matches in the regular expression, and have configured delays of three and five milliseconds, the system plays tone 1, delays 3 milliseconds, plays tone 2, delays 5 milliseconds, plays tone 3, delays 5 milliseconds. **Example: set post-tone-delays 3** There is no default tone delay.
introduction *filePath*	Specifies the path to a WAV file that plays at the introduction of a call (no audio is sent through until the introduction completes). Use the file-play-verify action to ensure that the recording is of a format supported by the AA-SBC device. **Example: set introduction /cxc_common/intro1.wav** There is no default setting.

Property name	Description
music-on-hold *filePath*	Specifies the path to a WAV file that plays (in a loop) while the call is on hold. Use the file-play-verify action to ensure that the recording is of a format supported by the AA-SBC device. **Example: set music-on-hold /cxc_common/hold1.wav** There is no default setting.
inactivity-timeout {enabled | disabled}	Specifies whether the system can timeout an anchored media session due to inactivity. See Media session maintenance for more information. If you enable this feature, you must set the length of the inactivity timer. See Setting time and time intervals for information on entry format requirements. **Example: set inactivity-timeout enabled 1800** The default setting is **disabled**. If set to **enabled**, the default timer setting is 3600 seconds. Or, enter a value greater than 60.
inactivity-style {session | per-call-leg}	Specifies which parties of a call must stop sending RTP before the system activates the inactivity timer (if **enabled**). When set to **session**, the system activates the timer when all parties stop sending RTP. When set to **per-call-leg**, if one party stops sending RTP, the system activates the inactivity timer. **Example: set inactivity-style per-call-leg** The default setting is **session**.
monitor *monitorGroupReference*	Associates a playback configuration with the call session. The playback function allows you to record SIP calls for playback on the AA-SBC Management System or a configured endpoint. Enter a pointer to a previously configured monitor-group object. **Example: set playback “vsp monitor-group callRecord”** There is no default setting.

Property name	Description
packet-marking {disable | tos *value*}	Enables or disables packet marking. Marking (tagging) a packet provides a quality of service (QOS) indicator, which routers along the path may act on. You could use packet marking, for example, to give priority to voice calls over other traffic. The system writes the value you enter to the TOS (or DiffServ) field of the IP header. **Example: set packet-marking tos 128** Enter a value between 0 and 255 in hexadecimal or decimal format. The default is **0xa0**.
rtp-stats {enabled | disabled}	Enables or disables the collection and logging of RTP and call quality statistics to the system database. Note that this property must be **enabled** to: • display Mean Opinion Score (MOS) or Quality of Service (QoS) statistics for a call at the AA-SBC Management System. • display RTP values in accounting files or databases that the system is writing to. **Example: set rtp-stats enabled** The default setting is **disabled**.

Property name	Description
rtcp {pass | drop | generate-only-if-required | generate-always} {true | false}	Specifies the handing and generation of RTCP packets in this SIP call session. When configuring this property, you set an action for call senders and a logging capability. Note that this property is not available when performing transcoding. See the Release Notes for more information. Set an action that defines how the system responds to RTCP packets it receives from call senders: • **pass**—transmits all packets. • **drop**—drops any packets. • generate-only-if-required—generates RTCP packets if it detects that the sender is not generating them. • **generate-always**—always generates packets, regardless of whether the sender did. Configure whether to log session statistics to the system database: • **true**—system writes RTCP statistics to the database, along with its own statistics. • **false**—system ignores statistics. **Example: set rtcp generate-only-if-required false** The default setting is an action of **pass** with a log setting of **false**.
mirror {enabled | disabled}	Specifies whether calls that match the defined policy are mirrored to other boxes in the cluster. To use this feature you must also set **mirror-media-settings** to true in the cluster object. **Example: set mirror disabled** The default setting is **enabled**.

Property name	**Description**
answer-media-loopback {enabled | disabled}	Sets whether the system answers a loopback call. When **enabled**, the system answers the call and generates RTP according to the negotiations. When **disabled**, the system allows the call to proceed according to the rest of the configuration. Endpoints can use these loopback calls to test the quality of the media transport, in accordance with the IETF draft-ietf-mmusic-media-loopback-07.txt. **Example: set auto-media-loopback enabled** The default setting is **disabled**.
tag-routing {enabled | disabled}	Specifies whether tag routing is in use for media. If routing tags are configured for an interface (using the ip **routing-tag** property), and that interface has media configured on it, the tags are only used when this property is **enabled**. See Tag-based route selection for more information. **Example: set tag-routing disabled** The default setting is **enabled**.
encode-auto-anchor-tag {true | false}	*Secondary property*. Specifies whether the content of the x-cxc-info field of the SDP is encoded or in clear text. This property is only applicable if the **anchor** property is set to **auto**. The x-cxc-info field contains the information necessary for the system to make an auto anchoring decision. If set to **true**, the field is base-64 encoded. If set to **false** it is sent in clear text. **Example: set encode-auto-anchor-tag false** The default setting is **true**.
transcode-balance-ptime {true | false}	*Secondary property.* Specifies whether the system uses signaling to attempt to “coax” the originating phone to send RTP packets at the rate of the destination phone. When transcoding, RTP packets may be arriving at departing at different rates (as determined by the CODEC in use with the phone). When **true**, the system sends the originating phone the ptime value (interval) in use by the destination phone. **Example: set transcode-balance-ptime false** The default setting is **true**.

Session configuration objects

Property name	Description
transcode-auto-release {true | false}	*Secondary property.* Specifies whether the system passes packets without transcoding, thereby releasing the transcode license for those sessions. This only applies if the system is set to auto anchor (with the anchor property set to auto) and transcode (using transcode-media-types property) and auto anchoring is required due to reachability issues between the source and destination. When this property is set to **true** in that situation, if the source and destination have the same set of CODECs, then the system passes the packets without transcoding. If **false**, the system transcodes the packets, costing the license two sessions for the duration. **Example: set transcode-balance-ptime false** The default setting is **true**.
decode-telephone-events {true | false}	*Secondary property.* Specifies whether the system should decode DTMF packets and inject them into the audio stream. This property may be used in cases where only one of the endpoints supports DTMF per *RFC 2833, RTP Payload for DTMF Digits, Telephony Tones and Telephony Signals.* If **false**, the default, the phone is responsible for inserting the DTMF into the audio stream. If **true**, the system decodes the DTMF packets and inserts the resulting audio into the stream. **Example: decode-telephone-events true** The default setting is **false**.

Property name	Description
repair-empty-codec-list {true | false}	*Secondary property.* Specifies whether the system repairs an SDP that contains an illegal media description. When set to **true**, the system inserts a default well-known CODEC into the SDP, allowing the phone to process the message. It also sets the port to 0 (if it is not already), disabling the media stream. When set to **false**, the system makes no changes to the SDP. This property only applies to illegal video and audio media types. The system inserts payload type 34 (H263) for video and payload type 0 (PCMU) for audio as the default payload types. **Example: set repair-empty-codec-list true** The default setting is **false**.
strip-blocked-stream {true | false}	*Secondary property*. Specifies whether the system removes the m= line from the SDP when all CODECs within that line are blocked. CODECs can be blocked by several mechanisms—media-type filtering, CODEC preferences, and stripping of unplayable or unverifiable CODECs. If the system blocks all CODECs in an m= line, it disables the stream. If this property is set to **true**, the system removes the entire m= line from the SDP. If set to **false**, the stream is disabled (by setting the port to zero) but remains in the SDP with some CODECs. This property is only applicable in messages that contain multiple media streams (e.g., audio and video). **Example: set strip-blocked-stream true** The default setting is **false**.
preserve-sdp-order {true | false}	*Secondary property*. Sets whether the system attempts to preserve the SDP attribute order. When set to **true**, the system makes a best attempt to preserve the order. When **false**, the default, the system uses the order native to its SDP parser. **Example: set preserve-sdp-order true** The default setting is **false**.

Property name	Description
handle-unknown-lines-in-sdp {strip | pass | error}	*Secondary property.* Specifies how the system handles errored lines that it receives in an SDP. If set to **strip**, the default, the system removes the offending lines and forwards the packet. If set to **pass**, the system forwards the packet untouched. If set to **error**, the system sends a 488 message (Not Acceptable) back to the sender and logs a message to the event log. **Example: set handle-unknown-lines-in-sdp error** The default setting is **strip**.
rtp-min-consecutive *packets*	*Secondary property.* Specifies the number of consecutive RTP packets the system must receive to establish an RTP source as valid. **Example: set rtp-min-consecutive 5** The default setting is **3** packets.
rtp-sequence-discontinuity {enabled | disabled}	*Secondary property.* Specifies whether the system monitors for, detects, and corrects RTP sequence number discontinuity. In some cases, a gateway may change the CODEC for a packet, but keep the same synchronization source (SSRC). If the resulting sequence numbers are discontinuous, it causes problems for SRTP processing. When this property is **enabled**, the system changes the SSRC if it detects sequence problems. When **disabled**, it does nothing. **Example: set rtp-sequence-discontinuity enabled** The default setting is **disabled**.

Property name	Description
rtp-splice{enabled | disabled}	*Secondary property.* Configures a mechanism for maintaining RTP parameters. When **enabled**, AA-SBC attempts to maintain RTP parameters when it injects DTMF (or other RTP audio) into the RTP stream. This may result in additional processing on each RTP packet after the normal audio stream is resumed, but it is required for interoperability with some endpoints because they do not recognize the DTMF when RTP parameters change (SSRC, sequence numbers, and timestamps). Note that the **rtp-stats** property must be enabled for this property to work. **Example: set rtp-splice enabled** The default setting is **disabled**.
combine-recording-fragments {enabled | disabled}	*Secondary property.* Specifies whether the system checks the state of the RTP recording file when a call ends. If **enabled**, when a AA-SBC-recorded call ends, the media master determines whether the RTP recording file is fragmented across the cluster. (Fragmenting can occur when a failover causes part of the recording to reside on one box and part on another.) If there are fragments, the system copies each to the master box and assembles the entire file. When **disabled**, the system bypasses fragment checking, which boosts performance. **Example: set combine-recording-fragments disabled** The default setting is **enabled**.
auto-anchor-consider-nat {enabled | disabled}	*Secondary property.* Specifies whether to disable a portion of the AA-SBC anchoring algorithm. This property is only applicable if you have the **anchor** property set to **auto**. Typically, if the system is forwarding a call from behind a NAT, it would anchor the media stream. You may set this property to **disabled** if, for example, you have a phone behind a NAT destined for a server that can do NAT traversal and you want to release the media stream. **Example: set auto-anchor-consider-nat disabled** The default setting is **enabled**.

Property name	Description
default-session-bandwidth *kbps*	*Secondary property.* Specifies an initial bandwidth value to use when calculating the bandwidth usage for each leg of a media session. The resulting value (in accumulation with all other session values) determines whether a server pool server has reached the configured **max-bandwidth** setting. Note that the bandwidth usage value is based not on the actual traffic on the wire, but on a calculation done by the AA-SBC device. The calculation uses the value associated with the first known CODEC identified in the SDP for a usage rate. If there is not a known CODEC, or the value has not yet been determined from the SDP, the system uses this setting. If you set this property to zero, the system treats media streams with no known codecs in the SDP as using zero bandwidth. (They do not count against the bandwidth limit for a given server.) **Example: set default-session-bandwidth 31** Enter a value between 0 and 10,000; the default setting is **87** kbps.
SIP-response-code-on-media-resource-alloc-failures *code*	*Secondary property.* Specifies the SIP response code to send when there is a media allocation failure. The failure may occur for one of two reasons—either media ports are not configured on the interface or no ports are available because all configured media ports are in use. **Example: set SIP-response-code-on-media-resource-alloc-failures 488** Enter a value from 300 to 699. The default setting is **488** (Service Unavailable).
attributeless-auto-anchor {enabled | disabled}	*Secondary property.* When enabled in conjunction with the anchor-mode=auto, the AA-SBC attempts to auto-anchor streams without additional NNOS-E attributes in the SDP. **Example: set attributeless-auto-anchor enabled.** The default setting is **disabled**.

Property name	Description
release-provisionally-anchored-media {true | false}	*Secondary property*. Release media resources that have been provisionally anchored. **Example: set release-provisionally-anchored-media true** The default setting is **false**.
report-last-timestamp {enabled | disabled}	When enabled, the NN260 reports the timestamp of the last received media packet. **Example: set report-last-timestamp enabled** The default setting is **disabled**.
monitor-rfc-2833 {enabled | disabled}	Specifies whether to have the AA-SBC change SSRC when it detects RTP sequence number discontinuity on active SSRC. **Example: set monitor-rfc-2833 enabled** The default setting is **disabled**.
release-provisionally-anchored-media {true | false}	Release media resources that have been provisionally anchored. When **media > anchor** is set to auto, the AA-SBC attempts to determine when anchoring resources can be released based on IP addresses and routing-tags of the communicating endpoints. When these determinations cannot be made, the media is deemed “provisionally anchored.” In releases previous to Release 3.5.5, provisionally allocated media was released by default. **Example**: **set release-provisionally-anchored-media true** The default setting is **false**.

nat-traversal

Purpose

Specifies whether symmetric Real-time Transport Protocol (RTP) is applied to the session. When a SIP call passes through far-end NAT to reach the call recipient, the complications of NAT (or a firewall) create problems for call connections.To address this, you can configure AA-SBC to run symmetric RTP.

RTP is a packet-based communication protocol that adds timing and sequence information to provide end-to-end network transport functions for applications transmitting real-time data, such as audio or video. With symmetric RTP on, AA-SBC sends return RTP messages based on the source IP address and UDP port in received RTP messages. NAT only modifies data in the IP header—the Session Description Protocol (SDP) payload is left unchanged. By using the source IP address and UDP port from the received RTP message, AA-SBC sends traffic back to the NAT device, instead of the untranslated addresses in the SDP message.

Syntax

```
config vsp default-session-config media nat-traversal
config vsp policies session-policies policy name rule name
   session-config media nat-traversal
config vsp dial-plan dial-prefix entryName session-config media
   nat-traversal
config vsp dial-plan route name session-config media nat-traversal
config vsp dial-plan source-route name session-config media
   nat-traversal
config vsp session-config-pool entry name media nat-traversal
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables the NAT traversal configuration on this AA-SBC device. When **enabled**, the system uses the settings configured here; when **disabled** the system uses the default NAT traversal settings. **Example: set admin disabled** The default NAT traversal setting is **enabled**.
symmetricRTP {true | false}	Specifies whether to use symmetric RTP for SIP call sessions. It specifies how the system learns the public address in preference to the private address (behind the firewall). When **true**, the system uses the address and port numbers that it learns from received RTP packets as the destination address for the endpoint. When set to **false**, the system uses the address and port numbers that it learns from the SDP message as the destination address for the endpoint. **Example: set symmetricRTP true** The default symmetric RTP setting is **false**.
asymmetric-rtp-address *ipAddress/ mask*	Specifies addresses that are not compatible with symmetric RTP. When an address that matches this property is received in an SDP message, the system disables symmetric RTP when sending RTP to that endpoint. This property is only necessary in certain situations (e.g., when using the BroadWorks Media Server), and only when the **symmetricRTP** property is set to true. **Example: set asymmetric-rtp-address 192.168.10.10/32** There is no default setting.

recording-policy

Purpose

Specifies whether to record calls, and if so, how to handle unsupported CODECs. Note that if you enable the record property, enabling recording of the SIP session, you must also have enabled the anchor property of the media object.

Playing recorded calls

If the record property is enabled, you can use CLI actions or the AA-SBC Management System to replay the SIP call session. To use the CLI, use the mix-session action to mix the files manually. You can also:

- play the file using the file-play action.
- play the session using the playback **session** action.

Handling unplayable CODECs

When AA-SBC receives the SDP message (typically in either an INVITE or 200 OK), it contains information from the originator on supported CODECs. These CODECs may or may not be mixable by the AA-SBC device. When the **strip-unplayable** property is **enabled**, AA-SBC removes any CODECs that it cannot mix from the list, and forwards the INVITE to the destination. When **disabled**, AA-SBC forwards the INVITE with the CODEC list unchanged. AA-SBC then records the entire contents of the call, but when mixed, any data sent with an unsupported CODEC results in silence.

Note: If AA-SBC records a call containing an unplayable CODEC, any archiving operation that involves that record (because the **include-mixed-media** property of the archiving operation is set to **true**) will fail.

Syntax

```
config vsp default-session-config media recording-policy
config vsp policies session-policies policy name rule name
   session-config media recording-policy
config vsp dial-plan dial-prefix entryName session-config media
   recording-policy
config vsp dial-plan route name session-config media recording-policy
config vsp dial-plan source-route name session-config media
   recording-policy
config vsp session-config-pool entry name media recording-policy
```

Properties

Property name	Description
record {enabled | disabled}	Enables or disables recording of the SIP session to the default directory on the system disk system. See Playing recorded calls for more information. **Example: set record enabled** The default setting is **disabled**.
strip-unplayable {enabled | disabled}	Specifies whether or not to strip CODECs that the system does not support. (This is only applicable if recording is enabled.) See Handling unplayable CODECs for more information, specifically about the risk to archiving operations if set to **disabled**. Use the file-play-verify action to ensure that a recording is of a format supported by the AA-SBC device. **Example: set strip-unplayable disabled** The default setting is **enabled**.

media-scanner-settings

Purpose

Configure media scanner settings. When media-scanner-settings are enabled, the media-scanner is started after the outgoing call connects and the pre-scan-time has elapsed. The media-scanner monitors the signal strength and duration of the received audio to divide it into intervals.

Syntax

```
config vsp default session-config media-scanner-settings
config vsp session-config-pool entry <name> media-scanner-settings
config vsp policies session-policies policy <name> rule <name>
   session-config media-scanner-settings
```

Properties

Property name	Description
admin [enabled | disabled]	Enables or disables the media scanner settings for play-file-broadcast. **Example: set admin enabled** The default setting is **disabled**.
pre-scan-time	The number of milliseconds to delay before invoking the media scanner for speaker detection. **Example: set pre-scan-time 35** Min: 0 / Max: 4294967295 The default setting is **20** msecs.
max-scan-time	The maximum number of milliseconds before canceling media scanning due to timeout. **Example: set max-scan-time 25000** Min: 0 / Max: 4294967295 The default setting is **30000** msecs.
low-threshold	Enter the quiet signal power threshold in dbs. **Example: set low-threshold -25** Min: -63 / Max: 3 The default setting is **-36**.
high-threshold	Enter the talk or tone signal power threshold in dbs. **Example: set high-threshold -25** Min: -63 / Max: 3 The default setting is **-36**.
low-long-duration	The number of milliseconds of detected quiet before declaring a long-pause. **Example: set low-long-duration 1500** Min: 0 / Max: 4294967295 The default setting is **2000**.

Property name	Description
high-long-duration	The number of milliseconds of detected talk or tone before declaring a long-talk or stable-tone. **Example: set high-long-duration 500** Min: 0 / Max: 4294967295 The default setting is **900**.
averaging-window	*Secondary property* The window of time used when calculating signal strength. **Example: set averaging-window 93** Min: 10 / Max: 1000 The default setting is **100**.
nominal-rounding-factor	*Secondary property.* The signal strength is rounded to the nearest multiple of the value you enter for this property. **Example: set nominal-rounding-factor 4** Min: 1 / Max: 25 The default setting is **2**.

transcoding-policy

Purpose

The transcoding policy object allows you to configure the transcoding policy for the AA-SBC. This includes defining the preferred codec as deduced from SDP offers and answers, adapting to match received codecs, and rewriting rfc-2833 headers when encoding audio.

Syntax

```
config vsp session-config pool entry media transcoding-policy
```

Properties

Property name	Description
media-types	The types of codecs that may be transcoded. These values are added to the SDP. The following is a list of codecs that may be transcoded: -pcma -pcmu -g7221 -g723 -g728 -g729 -g726-16 -g726-24 -g726-32 -g726-30 -gsm -gsm-amr -iLBC **Example: set media-types gsm**
most-preferred [true | false]	When true, the AA-SBC forces audio to only use the most preferred codec. **Example: set most-preferred true** The default setting is **false**.
symmetric-codec [true | false]	When true, the AA-SBC adapts and matches the correct codec when the endpoint has switched the “primary” codec. **Example: set symmetric-codec true** The default setting is **false**.
balance-ptime [true | false]	When true, the AA-SBC attempts to balance RTP packet times with the SDP. **Example: balance-ptime true** The default is **true**.

Property name	Description
auto-release [true | false]	When true, the AA-SBC attempts to release transcode resources when auto-anchoring is enabled. **Example: set auto-release false** The default is **true**.
block-unknown [true | false]	When true, the AA-SBC blocks unnegotiated packet types. **Example: set block-unknown true** The default is **false**.
decode-telephone-events [true | false]	When true, the AA-SBC decodes telephone-events into audio during transcoding when both sides do not support telephone-events. **Example: set decode-telephone-events true** The default is **false**.

periodic-announcement

Purpose

Specifies a WAV file that will be periodically inserted into a call. Use this to enter a recording tone or hold message. Use the file-play-verify action to ensure that the recording is of a format supported by the AA-SBC device.

Syntax

```
config vsp default-session-config media periodic-announcement
config vsp policies session-policies policy name rule name
   session-config media periodic-announcement
config vsp dial-plan dial-prefix entryName session-config media
   periodic-announcement
config vsp dial-plan route name session-config media
   periodic-announcement
config vsp dial-plan source-route name session-config media
   periodic-announcement
config vsp session-config-pool entry name media periodic-announcement
```

Properties

Property name	Description
file *filePath*	Specifies the location on the system of a WAV file containing the announcement. **Example: set file /cxc/recordings/announce1** There is no default location.
period *seconds*	Specifies the number of seconds the system waits between insertions of the announcement into the call. Note that the system starts its timer at the beginning of the announcement. The file starts playing every *period* number of seconds, regardless of the length of the recording. This means that you will hear the announcement continuously if the file, or specified **duration** (below), are longer than the specified **period**. **Example: set period 45** Enter a value from 10 to 3,600. The default value is **30** seconds.
duration *milliseconds*	Specifies, in milliseconds, how much of the specified recording to play. If you specify 0, the system plays the recording in its entirety. Use this with a value set to insert a tone file, and set the number of milliseconds that would not be too intrusive. **Example: set duration 500** The default duration is **0**.

media-verify-config

Purpose

Enables or disables media verification for RTP sessions, and sets SIP session termination based on RTP and alert messaging. The **media-verify-config** object allows you to verify RTP and RTCP media streams negotiated over SIP sessions. The settings verify that the media traffic passing through AA-SBC matches the negotiated and legal scheme for CODECs (coder/decoders) operating on SIP signals.

Syntax

```
config vsp media-verify-config
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables the current **media-verify-config** object. If **enabled**, the system uses these settings if this object is included in the session configuration media object. **Example: set admin enabled** The default setting is **enabled**.
terminate-session {enabled | disabled}	Enables or disables SIP session termination if a RTP media verification alert is generated. If **enabled**, the system terminates a media session completely upon the first media verification error (B2B mode only). **Example: set terminate-session enabled** The default setting is **enabled**.
alert-frequency *milliseconds*	Sets the number of milliseconds the system should wait between generation of media alert messages, which indicate a media verification error. **Example: set alert-frequency 2000** Enter a value from 10 to 10,000. The default setting is **1000** ms.

rtp

Purpose

Enables or disables verification of the Real-Time Protocol (RTP) header. AA-SBC uses the timing and sequence information in RTP to reassemble packets appropriately for real-time audio and video.

Syntax

```
config vsp media-verify-config rtp
```

Properties

Property name	Description
check-header {enabled | disabled}	Enables or disables verification of the RTP header using the negotiated and agreed media. **Example: set check-header disabled** The default setting is **enabled**.

codec

Purpose

Specifies the list of either additional codecs to allow and/or verify beyond AA-SBC internal list, or allows overriding the parameters of AA-SBC internal codecs. The AA-SBC codec name list is pre-populated with internal codecs by their official Session Description Protocol (SDP) tag. Enter a predefined SDP tag to open the **codec** object. (Type a question mark at the command line to see the list of predefined tags.) Or, to add a new codec, specify its SDP tag.

Syntax

```
config vsp media-verify-config rtp codec sdpTag
```

Properties

Property name	Description
payload-type {automatic | manual *hexValue*| none}	Specifies how the system calculates the RTP payload, either: • **automatic**—the system determines the payload based on the SDP rtpmap attribute, indicating the payload-type of the specified codec tag. For nearly all cases, this should be left as automatic • **manual**—Use this if the codec is always known to be constant or if the SIP client does not correctly use the rtpmap attribute. Specify the hexadecimal value that specifies the media payload type. • **none**—N/A. **Example: set payload-type automatic** The default setting is **automatic**.

Property name	Description
packet-size {automatic | manual [*minSize*] [*maxSize*] | none}	Specifies how the system calculates the RTP packet size, either: • **automatic**—the system determines the packet size based on the codec tag. For nearly all cases, this should be left as automatic as codecs use RFCs and other SDP parameters (number of channels, bit rate, sampling frequency, etc.) to determine the correct minimum and maximum sizes. • **manual**—Use this if a client does not specify a parameter correctly, or this is not an internal codec. Specify the minimum and maximum sizes in bytes. • **none**—disables packet-size checking. **Example: set packet-size manual 160 160** The default setting is **automatic**. If set to manual, the default minimum size is 100 bytes; the default maximum size is 100 bytes. (The packet must be 100 bytes.)
packet-rate {automatic | manual [*maxSize*] | none}	Specifies how the system calculates the maximum RTP packet rate, either: • **automatic**—the system determines the packet rate based on the codec tag. For nearly all cases, this should be left as automatic as codecs use RFCs and other SDP parameters (number of channels, bit rate, sampling frequency, etc.) to determine the correct maximum rate. • **manual**—Use this if a client does not specify a parameter correctly, or this is not an internal codec. Specify the maximum rate in packets per second. • **none**—disables packet-rate checking. **Example: set packet-rate manual 50** The default setting is **automatic**. If set to manual, the default size is 100 packets per second.

rtcp-header

Purpose

Enables or disables verification of the Real-Time Control Protocol (RTCP) header. AA-SBC uses RTCP to maintain quality of service and derive diagnostic data on RTP sessions.

Syntax

```
config vsp media-verify-config rtcp-header
```

Properties

Property name	Description
`header {enabled | disabled}`	Enables or disables verification of the RTCP header using the negotiated and agreed media. **Example:** set admin enabled The default setting is **enabled**.

call-monitoring

Purpose

Provides third-party conferencing, allowing a third-party participant, such as an emergency service endpoint, to be added to a call in progress. The third-party endpoint may or may not be registered with AA-SBC. Note that this feature differs from the monitor-group three-way calling feature. When using monitor groups, calls can only be listened to. Through this object a third-party can join a conversation.

Syntax

```
config vsp default-session-config media call-monitoring
config vsp policies session-policies policy name rule name
   session-config media call-monitoring
config vsp dial-plan dial-prefix entryName session-config media
   call-monitoring
config vsp dial-plan route name session-config media call-monitoring
config vsp dial-plan source-route name session-config media
   call-monitoring
```

```
config vsp session-config-pool entry name media call-monitoring
```

Properties

Property name	Description
admin {enabled | disabled}	Specifies whether the system initiates conference calling. When **enabled**, and there is a session configuration match, the system makes a call to the specified third party. Upon answering, the third party is conferenced into the active call. You must also enable the media anchor property to use this feature. **Example: set admin enabled** The default setting is **disabled**.
monitor-uri *uri*	Specifies the third party endpoint that is to be conferenced in to the active call. Note that only one call monitoring endpoint is supported per session configuration. **Example: set monitor-uri http://localEmerg.Services.net/monitor0** There is no default setting.

in-encryption

Purpose

Sets the parameters for inbound encrypted media sessions when AA-SBC is anchoring a call. This is the portion from the initiator to the AA-SBC device. With this object you set the encryption requirements on the call, and the encryption method used.

Syntax

```
config vsp default-session-config in-encryption
config vsp policies session-policies policy name rule name
   session-config in-encryption
config vsp dial-plan dial-prefix entryName session-config
   in-encryption
config vsp dial-plan route name session-config in-encryption
config vsp dial-plan source-route name session-config in-encryption
config vsp session-config-pool entry name in-encryption
```

Properties

Property name	Description
mode {none | pass-thru | allow | follow | offer | reoffer | require}	Specifies the encryption requirements on the incoming call (from endpoint to AA-SBC device). The method of encryption used is determined by the **type** property. Select one of the following: • **none**—the system disables the encryption put forth by the incoming endpoint (i.e, it responds “no” to the encryption portion of the authentication handshake.) If the outbound endpoint requires encryption, the call is dropped. • **pass-thru**—the system passes cryptographic parameters through the box and does not participate in RTP encryption. This method renders some advanced media services unusable (in particular, recording, announcements, transcoding, call monitoring, RTP stats, media verification, and RTCP generation). When using this option, set mode to **pass-thru** in both in-encryption and out-encryption. • **allow**—if the incoming endpoint offers encryption to the AA-SBC device, the system answers with it. If the endpoint does not offer encryption, the system does not answer with or initiate encryption. • **follow**—if the outbound endpoint offers encryption, the system offers encryption (the type set by policy) to the inbound endpoint. • **offer**—the system offers encryption to the inbound endpoint when the session is first established. *Continued*

Property name	Description
mode *continued*	• **reoffer**—the system offers encryption to the inbound endpoint whether the message is an INVITE or a REINVITE. This setting is most applicable when an endpoint issues a REINVITE, and encryption was not required with the original INVITE. In this case, the system will again offer encryption when forwarding the message. • **require**—the call must come in with encryption specified or the system drops it. **Example: set mode allow** The default setting is **none**.
type {Linksys | RFC-1889 | RFC-3711}	Sets the type of encryption on inbound sessions. In choosing a type, the system uses the encryption expected by that device or application. Choose one of the following, setting the system to: • **Linksys**—uses SIP INFO messages to exchange mini-certificates and exchange a symmetric key. The media encryption is similar to RFC-3711, but done with AES-128 Countermode and with HMAC MD5 authentication. See the linksys action for more certificate information. • **RFC-1889**—use encryption as defined in *RFC 1889, RTP: A Transport Protocol for Real-Time Applications*. This mode is used for compatibility with Windows Messenger and Microsoft Office Communicator, neither of which currently support RFC-3711 encryption. Instead, it uses a DES-CBC encryption of the entire UDP payload (including RTP headers) with no authentication. • **RFC-3711**—use encryption as defined in *RFC 3711, The Secure Real-time Transport Protocol (SRTP)*. **Example: set type Linksys** The default setting is **RFC-3711**.

Property name	Description
require-tls {false | true}	Specifies the requirements of the signaling protocol for the call inbound leg. It defines whether the system offers SRTP over a non-secure (TCP or UDP) signaling connection. The action of this property depends on the setting of the **mode** property. When this property is set to: • **true**—the system only offers encryption when talking to a TLS client. If TLS and SRTP are required (**mode** is set to **require**), the system fails calls going to TCP/UDP clients. If the mode property is set to **offer** or **follow**, the system forwards the call without SRTP. • **false**—the system offers SDP messages according to the mode setting without regard for the signaling transport. This allows keys to be exchanged in an insecure message. In most cases, this property does not need to be modified because the system does not consider the in-leg transport (only whether or not crypto was offered). However, set this property to true to ensure that the system does not offer crypto to a client on the in-leg that is not using TLS. **Example: set require-tls true** The default setting is **false**.
priority-AES-128-CM-HMAC-SHA1-32 *value*	*Secondary property*. Sets a preference for 32-bit SHA1 authentication tags on incoming calls. The system supports (offers) both 32- and 80-bit authentication tags on ingress. A value of 0 disables support for the 32-bit tag. To enable, set a value between 1 (most preferred) and 5. **Example: set priority-AES-128-CM-HMAC-SHA1-32 2** The default setting is **1**.

Property name	Description
priority-AES-128-CM-HMAC-SHA1-80 *value*	*Secondary property*. Sets a preference for 80-bit SHA1 authentication tags on incoming calls. The system supports (offers) both 32- and 80-bit authentication tags on ingress. A value of 0 disables support for the 80-bit tag. To enable, set a value between 1 (most preferred) and 5. **Example: set priority-AES-128-CM-HMAC-SHA1-80 1** The default setting is **2**.
mki-length *bytes*	*Secondary property*. Provides support for the optional Master Key Identifier bit defined in *RFC 3711, The Secure Real-time Transport Protocol (SRTP)*. The value specify sets the number of bytes in the MKI. The system then sends the negotiated identifier of that length indicating which master key to use for decryption with each SRTP packet. Note that the endpoint must support this option. **Example: set mki-length 1** Enter a value between 0 and 4; the default setting is **0**.
mikey-offer-location {session | media-descriptor}	*Secondary property.* Controls where in the SDP the system stores the MIKEY offer (the “a=key-mgmt:mikey” line) when it is made. If MIKEY is offered to the system, it puts the MIKEY answer in the location where the offer was located. Set the location as the media descriptor or session level. **Example: set mikey-offer-location media-descriptor** The default setting is **session**.
mikey-time-tolerance *seconds*	*Secondary property.* Controls where in the SDP the system stores the MIKEY offer (the “a=key-mgmt:mikey” line) when it is made. If MIKEY is offered to the system, it puts the MIKEY answer in the location where the offer was located. Set the location as the media descriptor or session level. **Example: set mikey-time-tolerance 90** The default setting is **60** seconds.

Session configuration objects

Property name	Description
symmetric-address-failure {enabled | disabled}	*Secondary property.* Specifies whether the system learns the source IP address from the RTP/RTCP packets, even if the packets fail decryption. When **enabled**, the first packet in a particular stream that fails SRTP decryption causes a DroppedPacket notification to be sent to the application with the address of the packet. The application treats this like a srcIPChanged notification. **Example: set symmetric-address-failure enabled** The default setting is **disabled**.
treat-as-secure	*This is a secondary property*. Specifies whether a proprietary security indicator is used on the SIP interface; either ST-secure or ST-insecure. This setting only operates on the X-Siemends-Call-Type header when MIKEY encryption is involved in pass-thru mode. The following are valid values: - Disabled—Sets ST-insecure - Enabled —Sets ST-secure - Auto—If SRTP is active on both sides of the call, the AA-SBC allows the X-Siemends-Call-Type header to pass unchanged. If SRTP is not active on other side of the call, the header is set to ST-insecure. The “auto” value sets the interface to trusted. **Example**: **set treat-as-secure enabled** The default setting is **disabled**.

out-encryption

Purpose

Sets the parameters for outbound encrypted media sessions when AA-SBC is anchoring the call. This is the portion from AA-SBC to the call recipient. With this object you set the encryption requirements on the call, and the encryption method used. Because AA-SBC does not always know on the outbound leg the encryption method expected by the recipient (because that recipient is not in the registry), you must manually set the type of encryption to offer.

Note about RFC-1889 encryption type

When using RFC 1889, Microsoft clients do not typically do encryption unless it is offered as mandatory in the SDP by one of the clients. If you want encryption on the outbound side, you must set the **mode** property to **require**.

Syntax

```
config vsp default-session-config out-encryption
config vsp policies session-policies policy name rule name
   session-config out-encryption
config vsp dial-plan dial-prefix entryName session-config
   out-encryption
config vsp dial-plan route name session-config out-encryption
config vsp dial-plan source-route name session-config out-encryption
config vsp session-config-pool entry name out-encryption
```

Properties

Property name	Description
mode {none | allow | follow | offer | reoffer | require}	Specifies the encryption requirements on the outgoing call (from system to endpoint). The method of encryption used is determined by the **type** property. Select one of the following: • **none**—the system disables the encryption put forth by the outbound endpoint (i.e., it responds “no” to the encryption portion of the authentication handshake.) If the inbound endpoint requires encryption, the call is dropped. • **pass-thru**—the system passes cryptographic parameters through the box and does not participate in RTP encryption. This method renders some advanced media services unusable (in particular, recording, announcements, transcoding, call monitoring, RTP stats, media verification, and RTCP generation). When using this option, set mode to **pass-thru** in both in-encryption and out-encryption. • **allow**—if the outgoing endpoint offers encryption to the AA-SBC device, the system answers with it. If the endpoint does not offer encryption, the system does not answer with or initiate encryption. • **follow**—if the inbound endpoint offers encryption, the system offers encryption (the type set by policy) to the outbound endpoint. • **offer**—the system offers encryption to the outbound endpoint when the session is first established. *continued*

Property name	Description
mode *continued*	• **reoffer**—the system offers encryption to the outbound endpoint whether the message is an INVITE or a REINVITE. This setting is most applicable when an endpoint issues a REINVITE, and encryption was not required with the original INVITE. In this case, the system will again offer encryption when forwarding the message. • **require**—the call must come in with encryption specified or the system drops it. **Example: set mode allow** The default setting is **none**.
type {Linksys | RFC-1889 | RFC-3711}	Sets the type of encryption on outbound sessions. In choosing a type, the system uses the encryption expected by that device or application. Choose one of the following, setting the system to: • **Linksys**—uses SIP INFO messages to exchange mini-certificates and exchange a symmetric key. The media encryption is similar to RFC-3711, but done with AES-128 Countermode and with HMAC MD5 authentication. See the linksys action for more certificate information. • **RFC-1889**—use encryption as defined in *RFC 1889, RTP: A Transport Protocol for Real-Time Applications*. This mode is used for compatibility with Windows Messenger and Microsoft Office Communicator, neither of which currently support RFC-3711 encryption. Instead, it uses a DES-CBC encryption of the entire UDP payload (including RTP headers) with no authentication. Note that to enable outbound encryption when using FRC 1889, you must set **mode** to **require**. • **RFC-3711**—use encryption as defined in *RFC 3711, The Secure Real-time Transport Protocol (SRTP)*. **Example: set type RFC-1899** The default setting is **RFC-3711**.

Property name	Description
require-tls {false | true}	Specifies the requirements of the signaling protocol for a call outbound leg. It defines whether the system offers SRTP over a non-secure (TCP or UDP) signaling connection. The action of this property depends on the setting of the **mode** property. When this property is set to: • **true**—the system only offers encryption when talking to a TLS client. If TLS and SRTP are required (**mode** is set to **require**), the system fails calls going to TCP/UDP clients. If the mode property is set to **offer** or **follow**, the system forwards the call without SRTP. • **false**—the system offers SDP messages according to the mode setting without regard for the signaling transport. This allows keys to be exchanged in an insecure message. Most phones follow *RFC 4568, SDP Security Descriptions for Media Streams*, and thus require that this property be set to **true**. **Example: set require-tls true** The default setting is **false**.
priority-AES-128-CM-HMAC-SHA1-32 *value*	*Secondary property*. Sets a preference for 32-bit SHA1 authentication tags on outgoing calls. By default, the system supports only 80-bit authentication tags on egress. Use this property to enable support for the 32-bit tag and set its preference relative to the 80-bit tag. To enable, set a value between 1 (most preferred) and 5. **Example: set priority-AES-128-CM-HMAC-SHA1-32 2** The default setting is **0** (disabled).

Property name	Description
priority-AES-128-CM-HMAC-SHA1-80 *value*	*Secondary property*. Sets a preference for 80-bit SHA1 authentication tags on outgoing calls. By default, the system supports (offers) only 80-bit authentication tags on egress. A value of 0 disables support for the 80-bit tag. To enable, set a value between 1 (most preferred) and 5. **Example: set priority-AES-128-CM-HMAC-SHA1-80 1** The default setting is **1**.
mki-length *bytes*	*Secondary property*. Provides support for the optional Master Key Identifier bit defined in *RFC 3711, The Secure Real-time Transport Protocol (SRTP)*. The value specify sets the number of bytes in the MKI. The system then sends the negotiated identifier of that length indicating which master key to use for decryption with each SRTP packet. Note that the endpoint must support this option. **Example: set mki-length 1** Enter a value between 0 and 4; the default setting is **0**.
mikey-offer-location {session | media-descriptor}	*Secondary property.* Controls where in the SDP the system stores the MIKEY offer (the “a=key-mgmt:mikey” line) when it is made. If MIKEY is offered to the system, it puts the MIKEY answer in the location where the offer was located. Set the location as the media descriptor or session level. **Example: set mikey-offer-location media-descriptor** The default setting is **session**.
mikey-time-tolerance *seconds*	*Secondary property.* Controls where in the SDP the system stores the MIKEY offer (the “a=key-mgmt:mikey” line) when it is made. If MIKEY is offered to the system, it puts the MIKEY answer in the location where the offer was located. Set the location as the media descriptor or session level. **Example: set mikey-time-tolerance 90** The default setting is **60** seconds.

Property name	Description
symmetric-address-failure {enabled | disabled}	*Secondary property.* Specifies whether the system learns the source IP address from the RTP/RTCP packets, even if the packets fail decryption. When **enabled**, the first packet in a particular stream that fails SRTP decryption causes a DroppedPacket notification to be sent to the application with the address of the packet. The application treats this like a srcIPChanged notification. **Example: set symmetric-address-failure enabled** The default setting is **disabled**.
treat-as-secure	*This is a secondary property*. Specifies whether a proprietary security indicator is used on the SIP interface; either ST-secure or ST-insecure. This setting only operates on the X-Siemends-Call-Type header when MIKEY encryption is involved in pass-thru mode. The following are valid values: - Disabled—Sets ST-insecure - Enabled —Sets ST-secure - Auto—If SRTP is active on both sides of the call, the AA-SBC allows the X-Siemends-Call-Type header to pass unchanged. If SRTP is not active on other side of the call, the header is set to ST-insecure. The “auto” value sets the interface to trusted. **Example: set treat-as-secure enabled** The default setting is **disabled**.

media-type

Purpose

Sets the media types that are allowed and/or prohibited during the session. You select a media type—audio, video, application, or MIME—and then a specific subtype. Use the question mark character at the command line to see a list of available subtypes. For example:

```
config media-type> set allowed-media-types audio ?

 allow sessions to use these media types

 syntax: set allowed-media-types audio sub-type
         set allowed-media-types video sub-type
         set allowed-media-types application sub-type
         set allowed-media-types custom-mime-type mime-type sub-type

 any
 pcmu
 gsm
 g723
 dvi4
 lpc
 pcma
 g722
--more--
```

In addition to the pre-configured options, you can allow or block custom types that may be part of your enterprise.

Syntax

```
config vsp default-session-config media-type
config vsp policies session-policies policy name rule name
   session-config media-type
config vsp dial-plan dial-prefix entryName session-config media-type
config vsp dial-plan route name session-config media-type
config vsp dial-plan source-route name session-config media-type
config vsp session-config-pool entry name media-type
```

Properties

Property name	Description
allowed-media-types {audio *subType* | video *subType* | application *subType* | custom-mime-type *mimeType subType*}	Sets the media types allowed during the session. Re-execute the command for each type you want to allow. **Example: set allowed-media-types custom-mime-type application safe-trade** The default subtype setting for audio, video, and application is **any**; there is no default setting for custom-mime-type.
blocked-media-types {audio *subType* | video *subType* | application *subType* | custom-mime-type *mimeType subType*}	Sets the media types to prohibit during the session. Re-execute the command for each type you want to block. **Example:** `set blocked-media-types video mpls` The default subtype setting for audio, video, and application is **any**; there is no default setting for custom-mime-type.

bodypart-type

Purpose

Sets the body types that are allowed and/or prohibited during the session. This functionality is initiated (if configured) when AA-SBC receives a SIP message that contains more than one type in the body portion of the message (when the Content Type header indicates that the message has multiple and mixed parts.) This object defines which types survive and which are deleted from the message.

Select a body type—application or text—and then a specific subtype. Use the custom MIME type to either add a body part other than application or text, or to use a subtype for application or text that is not preconfigured. To allow only a single body type, set **allowed-body-part** to the desired type and set **blocked-body-type** to **any**.

Syntax

```
config vsp default-session-config bodypart-type
config vsp policies session-policies policy name rule name
   session-config bodypart-type
config vsp dial-plan dial-prefix entryName session-config
   bodypart-type
config vsp dial-plan route name session-config bodypart-type
config vsp dial-plan source-route name session-config bodypart-type
config vsp session-config-pool entry name bodypart-type
```

Properties

Property name	Description
allowed-body-part {application *subType* | text *subType* | custom-mime-type *mimeType subType*}	Sets the body part types allowed during the session. Re-execute the command for each type you want to allow. **Example: set allowed-body-part custom-mime-type application safe-trade** The default subtype setting for audio, video, and application is **any**; there is no default setting for custom-mime-type.
blocked-body-part {application *subType* | text *subType* | custom-mime-type *mimeType subType*}	Sets the body part types to prohibit during the session. Any body sections that contain this type are removed from the message before forwarding. Re-execute the command for each type you want to block. **Example: set blocked-body-part application isup** The default subtype setting for audio, video, and application is **any**; there is no default setting for custom-mime-type.
move-bp-headers {enabled | disabled}	Specifies how to handle headers when there are changes to the message body. If **enabled**, when a message that has multiple parts is reduced into a single body part, the system moves the remaining body part header into the message header. **Example: set move-bp-headers enabled** The default setting is **disabled**.

dns-client-settings

Purpose

Configures the DNS client process and how the client communicates with the DNS service (resolver). See DNS service resolver and server objects for information on the AA-SBC DNS service.

Syntax

```
config vsp default-session-config dns-client-settings
config vsp policies session-policies policy name rule name
   session-config dns-client-settings
config vsp dial-plan dial-prefix entryName session-config
   dns-client-settings
config vsp dial-plan route name session-config dns-client-settings
config vsp dial-plan source-route name session-config
   dns-client-settings
config vsp session-config-pool entry name dns-client-settings
```

Properties

Property name	Description
admin {enabled | disabled}	Specifies whether this DNS client configuration entry is applied to calls matching the session configuration. **Example: set admin enabled** The default setting is **disabled**.
client-timeout *milliseconds*	Specifies how long the client process waits for the DNS service (resolver) to respond. If this timer expires, the service continues to look for the entry, and on finding it, writes it to its own and to the client cache. When the client next queries for that same address, the response will come from the cache. **Example: set client-timeout 3000** The default setting is **2000** milliseconds.
server-names *dnsServerReference*	Specifies which server(s) the DNS service should use to resolve requests that originate from calls matching this session configuration. Enter a reference to a server name that you configured with the resolver server object. If you do not configure any server-names with this property, all configured servers are used. **Example: set server-names "vsp dns resolver server name dns1"** There is no default setting.
routing-last-resort-dns {enabled | disabled}	Specifies whether the system should do a DNS lookup when it cannot determine where to forward a call based on the dial plan, registration plan, location cache, or policy. If **enabled**, the system does a DNS lookup on servers configured with the DNS resolver server object. If **disabled**, the system does not do a DNS lookup and returns, by default, a "404 not found" message to the caller. You can change the response code and string using the sip-settings **dns-fail-response-code** and **-string** properties. **Example: set routing-last-resort-dns enabled** The default setting is **disabled**.

Property name	Description
add-nnos-domain {enabled | disabled}	*Secondary property.* Specifies whether to add the configured domain to a single-label query. If disabled, no domain is added and the setting in the use-cxc-domain-in-search and additional-search-domains properties of the resolver object are applied. If enabled, AA-SBC appends the name set with the domain-name property of the static-stack-settings object to a single-label query, making it a FQDN. The other resolver settings do not apply. **Example: set add-nnos-domain disabled** The default setting is **enabled**.
`routing-lookup-type {NAPTR+SRV+A | NAPTR+SRV | SRV+A | SRV | A}`	*Secondary property.* Sets the method of server location. This property is applicable when the **routing-last-resort-dns** property is enabled. Select one of the following methods, in which the system performs: • **NAPTR+SRV+A**—a NAPTR lookup, an SRV lookup on the information returned, and an A lookup from those results. It also does a lookup on the original domain name. • **NAPTR+SRV**—a NAPTR lookup, an SRV lookup on the information returned, and an A lookup from those results. There is no lookup on the original domain name. • **SRV+A**—an SRV lookup, followed by an A lookup on names returned by SRV. Finally, it does a lookup on the original domain name. • **SRV**—an SRV lookup, followed by an A lookup on names returned by SRV. • **A**—a lookup on the original domain name. **Example: set routing-lookup-type SRV** The default setting is **NAPTR+SRV+A**.

Property name	Description
ras-settings *settings*	Sets the configuration for scenarios when AA-SBC is communicating with an external H.323 gatekeeper. (This property is only applicable if the **server-type** property is set to **h323-gatekeeper**.) When AA-SBC registers on behalf of a client, these settings allow the systems to exchange registration, admission, and status (RAS) messages. Set the following: • **registration TTL**—sets the frequency, in seconds, of the system putting forward reregistrations. The default is **3600** seconds • **registration retries**—sets the number of attempts the system makes to register a client. before abandoning the request. A value of zero allows unlimited retries. The default setting is **5** retries. *Continued*
ras-settings *settings* *Continued*	• **endpoint alias**—assigns a string to identify the system to the gatekeeper. This string should be configured on the gatekeeper as well so that it can recognize calls from AA-SBC. • **supported prefix**—sets the value for gatekeepers that need digits prepended to a number. This requirement would be set in the gatekeeper dial plan. Usually, this field is left blank. • **reregister on UNREGISTER**—sets whether the system tries to reregister a client after having received an UNREGISTER from the gatekeeper. If this is enabled, AA-SBC tries to reregister the client up to the number f times specified with the registration-retires field. The default setting is **disabled**. • **gatekeeper call routing**—sets whether the system provides support for gatekeeper-routed calls. When true, AA-SBC does provide that support. The default setting is **false**. **Example: set ras-settings 5200 8 nnos-e-1 gk1 1! enabled true** The default settings are indicated in each field description.

in-codec-preferences

Purpose

Sets a preference for CODECs, influencing the AA-SBC ordering of them in the SDC on the inbound leg of a call. AA-SBC removes those CODECs with a zero priority from the SDP. CODEC preferences do not cause AA-SBC to add a CODEC to the SDP, but to remove and/or reorder existing CODECs according to their priority. AA-SBC places a CODEC whose priority is not specified in its original order, just ahead of the known auxiliary CODECs (e.g., telephone-events). This is not a mechanism to add CODECs into an SDP, only to order those that are already there (via transcoding or from the original offer/answer).

Syntax

```
config vsp default-session-config in-codec-preferences
config vsp policies session-policies policy name rule name
   session-config in-codec-preferences
config vsp dial-plan dial-prefix entryName session-config
   in-codec-preferences
config vsp dial-plan route name session-config in-codec-preferences
config vsp dial-plan source-route name session-config
   in-codec-preferences
config vsp session-config-pool entry name in-codec-preferences
```

Properties

Property name	Description
preferences {audio | video} *codec priority*	Assigns a priority to a given CODEC for inbound audio or video sessions. Select: • **media type**—select for audio or video. The associated CODEC (subtype) is preferred according to the priority for that media type. • **codec**—the CODEC to which the priority applies. Use the question mark character at the command line to see a list of available CODECs, or enter any CODEC name. • **priority**—sets a preference for the CODEC. The lower the number, the more preferred the CODEC. Assigning a priority value of zero disables the CODEC for the session. The system removes these CODECs before sending the SDP offer or answer. **Example: set preferences video** There are no default settings. Enter a priority in the range of 0 through 100, and a subtype.

out-codec-preferences

Purpose

Sets a preference for CODECs, influencing the AA-SBC ordering of CODECs in the SDC on the outbound leg of a call. AA-SBC removes those CODECs with a zero priority from the SDP. CODEC preferences do not cause AA-SBC to add a CODEC to the SDP, but to remove and/or reorder existing CODECs according to their priority. AA-SBC places a CODEC whose priority is not specified in its original order, just ahead of the known auxiliary CODECs (e.g., telephone-events). This is not a mechanism to add CODECs into an SDP, only to order those that are already there (via transcoding or from the original offer/answer).

Syntax

```
config vsp default-session-config out-codec-preferences
config vsp policies session-policies policy name rule name
   session-config out-codec-preferences
config vsp dial-plan dial-prefix entryName session-config
   out-codec-preferences
config vsp dial-plan route name session-config out-codec-preferences
config vsp dial-plan source-route name session-config
   out-codec-preferences
config vsp session-config-pool entry name out-codec-preferences
```

Properties

Property name	Description
`preferences {audio | video}` *`codec priority`*	Assigns a priority to a given CODEC for outbound audio or video sessions. Select: • **media type**—select for audio or video. The associated CODEC (subtype) is preferred according to the priority for that media type. • **codec**—the CODEC to which the priority applies. Use the question mark character at the command line to see a list of well-known CODECs, or enter any string. • **priority**—sets a preference for the CODEC. The lower the number, the more preferred the CODEC. Assigning a priority value of zero disables the CODEC for the session. The system removes these CODECs before sending the SDP offer or answer. **Example: set preferences audio 0 any** There are no default settings. Enter a priority in the range of 0 through 100, and a subtype.

in-media-normalization

Purpose

Changes the media descriptor string (e.g., the CODEC for audio or video) in the SDP. Use this in cases where a client is unable to understand a variation in name of a CODEC/media descriptor. For example, G729 is sometimes transmitted as G729a. Inbound media normalization applies to the segment from the initiator to the AA-SBC device.

Syntax

```
config vsp default-session-config in-media-normalization
config vsp policies session-policies policy name rule name
   session-config in-media-normalization
config vsp dial-plan dial-prefix entryName session-config
   in-media-normalization
config vsp dial-plan route name session-config in-media-normalization
config vsp dial-plan source-route name session-config
   in-media-normalization
config vsp session-config-pool entry name in-media-normalization
```

Properties

Property name	Description
normalize {audio | video | application | image | custom-mime-type *mimeType*} *initialSubType alternateSubType*	Specifies, for a media type, how to normalize a CODEC/media descriptor name. The initial subtype is the type the system matches on and replaces. The alternate subtype is the type that the system then inserts in the SDP to replace the initial subtype. You can select a pre-configured type or enter a custom type. **Example: set normalize video g729 g729a** The default media type is **audio**. There is no default initial or alternate sub types.

out-media-normalization

Purpose

Changes the media descriptor string (e.g., the CODEC for audio or video) in the SDP. Use this in cases where a client is unable to understand a variation in name of a CODEC/media descriptor. For example, G729 is sometimes transmitted as G729a. Outbound media normalization applies to the segment from AA-SBC to the call recipient.

Syntax

```
config vsp default-session-config out-media-normalization
config vsp policies session-policies policy name rule name
   session-config out-media-normalization
config vsp dial-plan dial-prefix entryName session-config
   out-media-normalization
config vsp dial-plan route name session-config out-media-normalization
config vsp dial-plan source-route name session-config
   out-media-normalization
config vsp session-config-pool entry name out-media-normalization
```

Properties

Property name	Description
normalize {audio | video | application | image | custom-mime-type *mimeType*} *initialSubType alternateSubType*	Specifies, for a media type, how to normalize a CODEC/media descriptor name. The initial subtype is the type the system matches on and replaces. The alternate subtype is the type that the system then inserts in the SDP to replace the initial subtype. You can select a pre-configured type or enter a custom type. **Example: set normalize video g729 g729a** The default media type is **audio**. There is no default initial or alternate sub types.

in-hold-translation

Purpose

Configures the SDP hold attributes that are sent to an endpoint. The in-hold-translation object applies to SDP bodies sent to the endpoint that initiated the call. The out-hold-translation object applies to SDP bodies sent to the endpoint that initially received the call. When a call/stream is put on hold, the endpoint putting the call on hold sends an SDP offer with certain recognizable SDP characteristics—SDP connection information (c-line) and hold attributes that typically include “inactive” or “sendonly.” The endpoint that is being put on hold responds with an SDP answer acknowledging the hold. This is normally expressed by including a “recvonly” or “inactive” hold attribute. Not all servers recognize all SDP hold characteristics, so these objects can be used to configure the SDP hold characteristics sent to a given server. AA-SBC recognizes the following hold attributes (for offer or answer): sendrecv, sendonly, recvonly, and inactive. When changing or removing hold attributes, AA-SBC will remove or overwrite any of these attributes in the SDP.

For example, an endpoint being put on hold may responds to a server offer with “a=inactive” or “a=recvonly.” Some servers may interpret an SDP answer of “a=inactive” as “I’m not listening, do not send me music-on-hold.” If the endpoint putting the call on hold will play music-on-hold anyway, you may configure AA-SBC change the answer attribute to “a=recvonly.” In that way, when the server receives the SDP answer with the “a=recvonly,” it may still play the music-on-hold.

Syntax

```
config vsp default-session-config in-hold-translation
config vsp policies session-policies policy name rule name
   session-config in-hold-translation
config vsp dial-plan dial-prefix entryName session-config
   in-hold-translation
config vsp dial-plan route name session-config in-hold-translation
config vsp dial-plan source-route name session-config
   in-hold-translation
config vsp session-config-pool entry name in-hold-translation
```

Properties

Property name	Description
admin {enabled | disabled}	Specifies whether this hold translation entry is applied to calls matching the session configuration. **Example: set admin enabled** The default setting is **disabled**.
offer-address {pass | zero | non-zero}	Specifies how AA-SBC modifies the SDP connection information (c-line) sent in an SDP offer. Set the system to: • **pass**—not modify the c-line. • **zero**—change the address reported in the c-line to 0.0.0.0. • **non-zero**—change the address reported in the c-line to the last known address, typically the system interface address. **Example: set offer-address zero** The default setting is **pass**.

Property name	Description
offer-attribute {pass | remove | inactive | sendonly | sendrecv}	Specifies how AA-SBC modifies the SDP hold attributes in the SDP offer. Set the system to: • **pass**—not modify the hold attributes. • **remove**—remove any recognized hold attributes from SDP. • **inactive**—set the hold attribute to “inactive.” • **sendonly**—set the hold attribute to “sendonly.” • **sendrecv**—set the hold attribute to “sendrecv.” Use with care; this setting effectively tells the endpoint that the stream is not on hold. **Example: set offer-attribute inactive** The default setting is **pass**.
answer-address {pass | zero | non-zero}	Specifies how AA-SBC modifies the SDP connection information (c-line) sent in an SDP answer. Set the system to: • **pass**—not modify the c-line. • **zero**—change the address reported in the c-line to 0.0.0.0. • **non-zero**—change the address reported in the c-line to the last known address, typically the system interface address. **Example: set answer-address zero** The default setting is **pass**.

Property name	Description
answer-attribute {pass | remove | inactive | sendonly | sendrecv}	Specifies how AA-SBC modifies the SDP hold attributes in the SDP answer. Set the system to: • **pass**—not modify the hold attributes. • **remove**—remove any recognized hold attributes from SDP. • **inactive**—set the hold attribute to “inactive.” • **sendonly**—set the hold attribute to “sendonly.” • **sendrecv**—set the hold attribute to “sendrecv.” Use with care; this setting effectively tells the endpoint that the stream is not on hold. **Example: set answer-attribute remove** The default setting is **pass**.
remove-telephone-events {true | false}	*Secondary property*. Specifies whether the system strips telephone-events from the SDP when a call is placed on hold. When set to **true**, the system does strip events, which may be necessary for some phones (Polycom, for example). When **false**, the system does not modify events in the SDP. **Example: set remove-telephone-events true** The default setting is **false**.

out-hold-translation

Purpose

See the in-hold-translation object for a complete description.

Syntax

```
config vsp default-session-config out-hold-translation
config vsp policies session-policies policy name rule name
   session-config out-hold-translation
config vsp dial-plan dial-prefix entryName session-config
   out-hold-translation
config vsp dial-plan route name session-config out-hold-translation
config vsp dial-plan source-route name session-config
   out-hold-translation
config vsp session-config-pool entry name out-hold-translation
```

in-dtmf-translation

Purpose

Controls the method used for forwarding DTMF tones in a call. The two supported methods are via the signaling stream using SIP INFO messages or via a DTMF packet that is in compliance with RFC 2833, RTP Payload for DTMF Digits, Telephony Tones and Telephony Signals. Using this object, you can configure AA-SBC to:

- forward packets in the form they arrived.
- pick them out of the RTP stream and send them in a SIP INFO message (if they arrived as DTMF packets).
- extract them from a SIP INFO message and send them as DTMF packets (if the SIP INFO message contained a DTMF body).

Inbound DTMF translation applies to the segment from the initiator to the AA-SBC device. See *Net-Net OS-E – Session Services Configuration Guide* for detailed information on DTMF.

Syntax

```
config vsp default-session-config in-dtmf-translation
config vsp policies session-policies policy name rule name
   session-config in-dtmf-translation
config vsp dial-plan dial-prefix entryName session-config
   in-dtmf-translation
config vsp dial-plan route name session-config in-dtmf-translation
config vsp dial-plan source-route name session-config
   in-dtmf-translation
config vsp session-config-pool entry name in-dtmf-translation
```

Properties

Property name	Description
info {info | rfc-2833}	Specifies the method to use for forwarding DTMF tones that were received in a SIP INFO message. If set to **info**, the system forwards the message as it was received (in an INFO message). If set to **rfc-2833**, the system extracts the DTMF body from the INFO message and sends the content via DTMF packets. **Example: set info rfc-2833** The default setting is **info**.
drop-info {true | false}	Specifies whether to drop the SIP INFO message if the info property is set to rfc-2833. If set to **true**, the system drops the INFO packet and only sends the DTMF packets. If set to **false**, the system sends both. **Example: set drop-info true** The default setting is **false**.
rfc-2833 {info | rfc-2833}	Specifies the method to use for forwarding DTMF tones that were received in DTMF packets. If set to **rfc-2833**, the system forwards the message as it was received (in DTMF packets). If set to **info**, the system extracts the DTMF packets from the RTP stream and sends the content via a SIP INFO message. The system sends one INFO message per event detected. **Example: set rfc-2833 info** The default setting is **rfc-2833**.

Property name	Description
drop-rfc-2833 {true | false}	Specifies whether to drop the DTMF packets if the rfc-2833 property is set to info. If set to **true**, the system drops the RFC 2833 packets and only sends the SIP INFO message. If set to **false**, the system sends both. **Example: set drop-rfc-2833 true** The default setting is **false**.
info-dtmf-body {dtmf-relay | dtmf}	Specifies the body type to use in a SIP INFO message when converting from RFC 2833 format. When using **dtmf**, the message body contains just single character (the digit that was pressed). When set to **dtmf-relay**, the body contains the single character plus duration data. **Example: set info-dtmf-body dtmf** The default setting is **dtmf-relay**.
timeout-rfc-2833 *milliseconds*	*Secondary property*. Sets the number of milliseconds the system waits before sending a SIP INFO message if it does not detect the end of the event. The timer is started at the start of an event. This property only applies when the forwarding method has been changed from **rfc-2833** to **info**, and is used in the event that when monitoring DTMF, the system does not detect an event end. **Example: set timeout-rfc-2388 1500** The default setting is **1000** milliseconds.

out-dtmf-translation

Purpose

Controls the method used for forwarding DTMF tones in a call. The two supported methods are via the signaling stream using SIP INFO messages or via a DTMF packet that is in compliance with RFC 2833, RTP Payload for DTMF Digits, Telephony Tones and Telephony Signals. Using this object, you can configure AA-SBC to:

- forward packets in the form they arrived.
- pick them out of the RTP stream and send them in a SIP INFO message (if they arrived as DTMF packets).
- extract them from a SIP INFO message and send them as DTMF packets (if the SIP INFO message contained a DTMF body).

Outbound DTMF translation applies to the segment from AA-SBC to the call recipient. See *Net-Net OS-E – Session Services Configuration Guide* for detailed information on DTMF.

Syntax

```
config vsp default-session-config out-dtmf-translation
config vsp policies session-policies policy name rule name
   session-config out-dtmf-translation
config vsp dial-plan dial-prefix entryName session-config
   out-dtmf-translation
config vsp dial-plan route name session-config out-dtmf-translation
config vsp dial-plan source-route name session-config
   out-dtmf-translation
config vsp session-config-pool entry name out-dtmf-translation
```

Properties

Property name	Description
info {info | rfc-2833}	Specifies the method to use for forwarding DTMF tones that were received in a SIP INFO message. If set to **info**, the system forwards the message as it was received (in an INFO message). If set to **rfc-2833**, the system extracts the DTMF body from the INFO message and sends the content via DTMF packets. **Example: set info rfc-2833** The default setting is **info**.
drop-info {true | false}	Specifies whether to drop the SIP INFO message if the info property is set to rfc-2833. If set to **true**, the system drops the INFO packet and only sends the DTMF packets. If set to **false**, the system sends both. **Example: set drop-info true** The default setting is **false**.
rfc-2833 {info | rfc-2833}	Specifies the method to use for forwarding DTMF tones that were received in DTMF packets. If set to **rfc-2833**, the system forwards the message as it was received (in DTMF packets). If set to **info**, the system extracts the DTMF packets from the RTP stream and sends the content via a SIP INFO message. The system sends one INFO message per event detected. **Example: set rfc-2833 info** The default setting is **rfc-2833**.
drop-rfc-2833 {true | false}	Specifies whether to drop the DTMF packets if the rfc-2833 property is set to info. If set to **true**, the system drops the RFC 2833 packets and only sends the SIP INFO message. If set to **false**, the system sends both. **Example: set drop-rfc-2833 true** The default setting is **false**.

Property name	Description
info-dtmf-body {dtmf-relay | dtmf}	Specifies the body type to use in a SIP INFO message when converting from RFC 2833 format. When using **dtmf**, the message body contains just single character (the digit that was pressed). When set to **dtmf-relay**, the body contains the single character plus duration data. **Example: set info-dtmf-body dtmf** The default setting is **dtmf-relay**.
timeout-rfc-2833 *milliseconds*	*Secondary property*. Sets the number of milliseconds the system waits before sending a SIP INFO message if it does not detect the end of the event. The timer is started at the start of an event. This property only applies when the forwarding method has been changed from **rfc-2833** to **info**, and is used in the event that when monitoring DTMF, the system does not detect an event end. **Example: set timeout-rfc-2388 1500** The default setting is **1000** milliseconds.

sdp-regeneration

Purpose

Sets parameters to “regenerate” the SDP in order to more tightly control what is sent out by AA-SBC. This ensures that approved SDP format comes from the system on every call.

Manipulating connection information

Some phones may require configuration ensuring that connection information is not specified within the media descriptor. Using the **add-session-connection** and **remove-media-connection** properties together, you can configure AA-SBC to add a session-level c-line and remove matching c-lines from the media descriptor.

The SDP c-line, which contains connection data for the session, can be found in each media description and/or at the session level. If it appears at both the session level and in the media descriptor, the media descriptor value takes precedence (for that descriptor).

Syntax

```
config vsp default-session-config sdp-regeneration
config vsp policies session-policies policy name rule name
   session-config sdp-regeneration
config vsp dial-plan dial-prefix entryName session-config
   sdp-regeneration
config vsp dial-plan route name session-config sdp-regeneration
config vsp dial-plan source-route name session-config sdp-regeneration
config vsp session-config-pool entry name sdp-regeneration
```

Properties

Property name	Description
regenerate {enabled | disabled}	Controls whether the system applies the settings in this object. When **enabled**, all settings are applied to the SDP and the system regenerates it before passing it on. **Example: set regenerate disabled** The default setting is **enabled**.
origin {pass | rewrite}	Specifies whether the system overwrites the string that appears in the origin line (o=) of the SDP. If set to **rewrite**, the system changes the value of the username to CSM or the value set with the **username** property. In addition, it changes the value of the session-id and session-version to zero. Otherwise, it passes the name unchanged. **Example: set origin pass** The default setting is **rewrite**.
username *string*	Specifies the username to be inserted into the username field of the origin line of the SDP. This value is only applicable if the origin property is set to **rewrite**. **Example: set username joe@cov.com** There is no default setting.

Property name	Description
session-name {pass | rewrite}	Specifies whether the system overwrites the textual session name that appears in the session-name (s=) line of the SDP. If set to **rewrite**, the system changes the content of the session name to the value set with the **name** property. Otherwise, it passes the session-name unchanged. **Example: set session-name pass** The default setting is **rewrite**.
name *string*	Specifies the name to be inserted into the session-name of the SDP. This value is only applicable if the origin property is set to **rewrite**. **Example: set name “multimedia conference”** There is no default setting.
session-info {pass | strip}	Specifies whether to strip out or pass the textual information in the session-info (i=) line of the SDP. **Example: set session-info pass** The default setting is **strip**.
uri {pass | strip}	Specifies whether to strip out or pass the uri (u=) line of the SDP, a pointer to additional information about the session. **Example: set uri pass** The default setting is **strip**.
e-mail-address {pass | strip}	Specifies whether to strip out or pass the email contact information for the person responsible for the conference. This is displayed in the e= line of the SDP. **Example: set e-mail-address pass** The default setting is **strip**.
phone-number {pass | strip}	Specifies whether to strip out or pass the telephone contact information for the person responsible for the conference. This is displayed in the p= line of the SDP. **Example: set phone-number pass** The default setting is **strip**.

Property name	Description
bandwidth {pass | strip}	Specifies whether to strip out or pass the proposed bandwidth to be used by the session. This is displayed in the b= line of the SDP. **Example: set bandwidth pass** The default setting is **strip**.
timing {pass | strip}	Specifies whether to strip out or pass the start and stop times for a session. This is displayed in the t= line of the SDP. Note that some phones require a t-line for proper operation. **Example: set timing strip** The default setting is **pass**.
remove-unknown {enabled | disabled}	Specifies whether to remove any unknown lines from the SDP. When **enabled**, all unknown (non-specification) lines are removed. **Example: set remove-unknown disabled** The default setting is **enabled**.
add-session-connection {enabled | disabled}	Specifies whether to add session-level c-line content to the SDP if it is not already there. When **enabled**, the system inserts a session-level c-line (prior to the first m-line) in the SDP text message. The content for the line is derived from the first media descriptor c-line. Note that this property only adds the c-line at the session level. To remove c-lines from the media-descriptors, you must use the **remove-media-connection** property. See Manipulating connection information for more information. **Example: set add-session-connection enabled** The default setting is **disabled**.
remove-media-connection {enabled | disabled}	Specifies whether to remove the c-line content from the SDP media descriptors. When **enabled**, the system removes all c-lines within media descriptors that match the session-level c-line. See Manipulating connection information for more information. **Example: set remove-media-connection enabled** The default setting is **disabled**.

Property name	Description
add-rtpmaps {enabled | disabled}	Specifies whether the system includes the rtpmap attributes for well-known CODECs (that is “knows” about), when the rtpmap is not included by the original endpoint. Payload types under 96 must be registered with IANA as well-known CODECs. An rtpmap attribute for well-known codecs is not required in the SDP and, therefore, not included by some endpoints. However, certain endpoints have processing problems when the rtpmap is not included. When **enabled**, the system adds the rtpmap attributes. **Example: set add-rtpmaps enabled** The default setting is **disabled**.
pass-attribute {*string* | *aLine*}	Identifies specific attribute lines to be passed through the system unchanged. Enter as many attributes as you require. Select from a predefined list or enter the attribute name. By default the system always passes certain attributes having to do with call flow (e.g., a=sendrecv) and cryptography (e.g., a=key-mgmt, a=crypto). **Example: set pass-attribute rtcp** The default setting is **disabled**.

sip-directive

Purpose

Sets the default instruction to apply to SIP packets arriving to open a new session. AA-SBC can either allow the session to come up, refuse it (but send a polite error response), or discard the SIP message that started the session and ignore the request.Use the SIP directive in the default session config object to apply instructions for calls in which there are no configured policies or rules; use the policy session config object to define the action for calls that match policy.

Syntax

```
config vsp default-session-config sip-directive
config vsp policies session-policies policy name rule name
   session-config sip-directive
config vsp dial-plan dial-prefix entryName session-config
   sip-directive
config vsp dial-plan route name session-config sip-directive
config vsp dial-plan source-route name session-config sip-directive
config vsp session-config-pool entry name sip-directive
```

Properties

Property name	Description
directive {allow | discard | refuse *resultCode textString*}	Sets the default SIP call directive (instruction) to apply to the SIP call session, either: • **allow**—the system allows the packet through. • **discard**—the system immediately discards the packet. • **refuse**— the system discards the packet but sends a response to indicate having done so. Optionally you can specify a SIP code between 400 and 699 and a text message to be appended to a refused SIP call record. If set to the **refuse** option, the code string is not visible. **Example: set directive refuse 401 “Server not available.”** The default setting is **discard**.

inbound-request-uri-specification

Purpose

Specifies whether AA-SBC modifies the content of the host, port, and/or transport fields of the REQUEST URI. If set, changes are applied only to the REQUEST message traveling in the opposite direction of the session initiation REQUEST message. (Use the request-uri-specification object to change outbound REQUEST URIs.) These properties should only be changed to override the default behavior because of issues with an intermediary device.

Syntax

```
config vsp default-session-config inbound-request-uri-specification
config vsp policies session-policies policy name rule name
   session-config inbound-request-uri-specification
config vsp dial-plan dial-prefix entryName session-config
   inbound-request-uri-specification
config vsp dial-plan route name session-config
   inbound-request-uri-specification
config vsp dial-plan source-route name session-config
   inbound-request-uri-specification
config vsp session-config-pool entry name
   inbound-request-uri-specification
```

Properties

Property name	Description
host-use-next-hop {enabled | disabled}	Specifies whether to change the host portion of the REQUEST URI. If **enabled**, the system sets the host portion to the IP address of the next hop. If **disabled**, the host portion remains unchanged. **Example: set host-use-next-hop enabled** The default setting is **disabled**.

Property name	Description
port-use-next-hop {enabled | disabled}	Specifies whether to change the port specified in the REQUEST URI. If **enabled**, the system sets the port number to the port used for the next hop. If **disabled**, the port number remains unchanged. **Example: set port-use-next-hop enabled** The default setting is **disabled**.
transport-use-next-hop {enabled | disabled}	Specifies whether to change the transport protocol specified in the REQUEST URI. If **enabled**, the system sets the transport to the protocol used by the next hop. If **disabled**, the transport protocol remains unchanged. **Example: set transport-use-next-hop enabled** The default setting is **disabled**.

contact-uri-settings-3xx-response

Purpose

Specifies where AA-SBC derives the content of the CONTACT header in 3xx (response/redirect) messages that it receives. The CONTACT headers in a 3xx response tell the recipient of the request (which caused the redirect to be generated) where the request can be sent that will yield a final response. This “alternate location” information is sent back to the UAC via the Contact: header(s) in the 3xx response. Set this object in cases when you want AA-SBC to forward the 3xx message with a CONTACT header containing something other than AA-SBC itself. (Note, however, that in that case, the UA does not return the updated information to the AA-SBC device.)

For example, if the user property is set to **to-uri**, AA-SBC replaces the user field of the CONTACT header with data from the user field of the outgoing TO header in the 3xx response.

Syntax

```
config vsp default-session-config contact-uri-settings-3xx-response
config vsp policies session-policies policy name rule name
   session-config contact-uri-settings-3xx-response
```

```
config vsp dial-plan dial-prefix entryName session-config
   contact-uri-settings-3xx-response
config vsp dial-plan route name session-config
   contact-uri-settings-3xx-response
config vsp dial-plan source-route name session-config
   contact-uri-settings-3xx-response
config vsp session-config-pool entry name
   contact-uri-settings-3xx-response
```

Properties

Property name	Description
user {request-uri | to-uri | from-uri | contact-uri | omit | *string*}	Specifies how to derive the value of the User field (the resource located at host) of the CONTACT header. • **request-uri**—uses the value from the incoming REQUEST URI. • **to-uri**—uses the value from the incoming TO URI. • **from-uri**—uses the value from the incoming FROM URI. • **next-hop**—uses the IP address of the next-hop server. • **omit**—leaves the field blank. • **string**—writes the specified string to the field. **Example: set user from-uri** The default setting is **contact-uri**.
user-prefix *string*	Appends the specified string to the beginning of the User field of the CONTACT header. Use this, for example, if you need to append a “1” to a phone number for an outside call. **Example: set user-prefix 1** There is no default setting.

Property name	Description
host {cxc-address | public-address | original-address | next-hop-address | *string*}	Specifies how to derive the value of the Host field (the host providing SIP resource) of the CONTACT header. • **cxc-address**—uses the AA-SBC IP address as the host • **public-address**—uses the public address for a UAC behind a firewall or the UAC address if it is not behind a firewall. • **original-address**—the host field is not modified. • **next-hop-address**—uses the IP address of the next-hop server. However, this value is typically used only for the • **string**—writes the specified string to the field. **Example: set host original-address** The default setting is **next-hop-address**.
port {cxc-local-port | original-port | omit | *string*}	Specifies how to derive the value of the Port field (where the request is to be sent) of the CONTACT header. • **cxc-local-port**—uses the port number that the system transported the call over. • **original-port**—the port field is not modified. • **omit**—leaves the field blank. • **string**—writes the specified string to the Port field of the 3xx CONTACT header. **Example: set port original-port** The default setting is **omit**.

Property name	Description
transport {next-hop-transport | original-transport | omit | UDP | TCP | TLS}	Specifies the derivation of the transport type for the Transport field of the CONTACT header. • **next-hop-transport**—uses the method used by the next-hop server. • **original-transport**—the transport field is not modified. • **omit**—leaves the field blank. • **UDP, TCP, TLS**—sets the transport field to the selected protocol. **Example: set transport original-transport** The default setting is **omit**.
add-maddr [enabled | disabled]	When enabled, the AA-SBC adds a maddr URI parameter if the original host is a fully qualified domain name (FQDN). Example: set add maddr enabled The default setting is **disabled**.

emergency-settings

Purpose

Sets whether matching calls should be handled without limitation. When this object is administratively enabled, matching calls are not subject to emission and admission controls.

Syntax

```
config vsp default-session-config emergency-settings
config vsp policies session-policies policy name rule name
   session-config emergency-settings
config vsp dial-plan dial-prefix entryName session-config
   emergency-settings
config vsp dial-plan route name session-config emergency-settings
config vsp dial-plan source-route name session-config
   emergency-settings
config vsp session-config-pool entry name emergency-settings
```

Session configuration objects

Properties

Property name	Description
admin {enabled | disabled}	Specifies whether the emergency setting should be active or inactive. When **enabled**, a matching call is handled without limitation (not subject to emission and admission controls). **Example: set emergency-settings enabled** The default setting is **disabled**.

calling-group-settings

Purpose

Specifies how matching calls are mapped to calling groups and the group reference from which they pick up their provisioning. See Chapter 11, “Calling Group objects” for complete information on calling groups.

Syntax

```
config vsp default-session-config calling-group-settings
config vsp policies session-policies policy name rule name
   session-config calling-group-settings
config vsp dial-plan dial-prefix entryName session-config
   calling-group-settings
config vsp dial-plan route name session-config calling-group-settings
config vsp dial-plan source-route name session-config
   calling-group-settings
config vsp session-config-pool entry name calling-group-settings
```

Properties

Property name	Description
type {dynamic | configured}	Sets the calling-group association. Select one of the following: • dynamic—creates a calling group, and names it based on the AOR of the device. The new group then inherits the settings of the group referenced in the **calling-group** property. • configured—assigns the matching AOR to the group referenced with the **calling-group** property. This assignment overrides any other calling group assignment (e.g., registration-plan route match). **Example: set type dynamic** There is no default setting.
calling-group *callingGroupReference*	Specifies the referenced group for calling group parameters. If **type** is set to configured, it is the group that the AOR joins. If set to dynamic, it is the group from which the new groups settings are inherited. **Example: set calling-group “vsp calling-groups groupEast”** There is no default setting.

presence

Purpose

Configures presence services for this SIP call session. SIP call messages containing SUBSCRIBE and NOTIFY requests and responses are associated with a presence session.

AA-SBC stores presence information in its local databases. Presence information pertains to your SIP presence (online, away, etc.). This object enables presence services and controls whether AA-SBC translates that presence information from one server type to another (because every server type has its own way of storing/ transmitting this data).

Syntax

```
config vsp default-session-config presence
config vsp policies session-policies policy name rule name
   session-config presence
config vsp dial-plan dial-prefix entryName session-config presence
config vsp dial-plan route name session-config presence
config vsp dial-plan source-route name session-config presence
config vsp session-config-pool entry name presence
```

Properties

Property name	Description
presence-services {enabled | disabled}	Enables or disables system presence services on this SIP call session. If set to **enabled**, the session information is sent to the system presence database. If **disabled**, the system allows packets through but does not write any information to the database. **Example: set presence-services disabled** The default setting is **disabled**.
presence-translation {disabled | sametime-to-lcs | lcs-to-sametime | lcs-to-lcs | st-to-st | mcs-to-mcs | voice-to-voice | voice-to-lcs}	Sets the server types, and therefore the translation direction, that the system performs. The system modifies an incoming request so that the far-end client receives presence data in a recognizable format. Select one of the following: • **disabled**—the request is passed through unmodified • **sametime-to-lcs**—provides Sametime-to-LCS translation • **lcs-to-sametime**—provides LCS-to-Sametime translation • **lcs-to-lcs**—ensures LCS-to-LCS presence data compatibility. • **st-to-st**—ensures Sametime-to-Sametime presence data compatibility. This would be for use, for example, when using different versions of Sametime. • **mcs-to-mcs**—ensures Nortel MCS-to-MCS presence data compatibility, for example, when different versions are running. • **voice-to-voice**—indicates that the entity is a phone/ voice (to support phones that want to SUBSCRIBE to a SIP server and announce presence). Identifying the entity as voice prevents the system from altering the SUBSCRIBE message in the presence database. • **voice-to-lcs**—enables forwarding of phone registrations to LCS (mostly useful in CSTA cases) to control OC client presence information. **Example: set presence-translation sametime-to-lcs** The default setting is **disabled**.

Property name	**Description**
presence-mapping *`value mapTo`*	Configures, for a given presence status, the value to map that status to. Use this property if both clients do not recognize the same presence states. In an LCS-to-Sametime configuration, for example, if status is set to a value of “Out to Lunch” with the LCS client, it must be mapped to an option that Sametime recognizes. Value and map options: • offline • away • out-to-lunch • on-the-phone • be-right-back • busy • do-not-disturb • online **Example: set presence-mapping out-to-lunch away** There is no default setting.
voice-lcs-transport {any | UDP | TCP | TLS}	Sets the default protocol used to transmit voice-to-LCS SIP packets to the destination LCS server. **Example: set voice-lcs-transport TLS** The default transport is **any**.
federation-contact *string*	Specifies the string that the system uses to create the SIP header URIs it sends to remote servers. The string must match the content of the Common Name field that is present in the certificate that the system provides to the LCS to prove its authenticity. **Example: set federation-contact companyABC.com** There is no default setting.

Property name	Description
message-body-replace *regExp replacement*	Alters the body of any SIP message (for example, IM content or the SDP for an INVITE) for a matching session. Use this property with caution; you would only change the SIP message body under specific required circumstances. In the example below, the system replaces sip:1002@company.com in the SIP message body with sip:9788231002@company.com. **Example: set message-body-replace “(.*)sip:(\d{4})@company.com(.*)” “\1sip:978823\2@company.com\3”** There is no default setting. For more information regarding configuring regular expressions and replacement strings, see the section in Chapter 1, “Using regular expressions” on page 36.
st-keep-alives {enabled | disabled}	Specifies whether the system responds to Sametime keep alives. When **enabled**, the system responds to keep alives on behalf of the federated domain, letting the Sametime server know that the remote peer is active. Use this in federated configurations using a version of Sametime that requires SIP keep alives. Otherwise, this property should be **disabled**. **Example: set st-keep-alives enabled** The default setting is **disabled**.
location-presence-service {enabled | disabled}	Specifies whether AA-SBC publishes device presence information to the jtapi master service. For Technical Support use only. **Example: set location-presence-service enabled** The default setting is **disabled**.

registration

Purpose

Configures properties that are used in processing REGISTER requests. Note that for AA-SBC to accept REGISTER requests, you must enable the **admin** property of the registration-service object.

Syntax

```
config vsp default-session-config registration
config vsp policies session-policies policy name rule name
   session-config registration
config vsp dial-plan dial-prefix entryName session-config registration
config vsp dial-plan route name session-config registration
config vsp dial-plan source-route name session-config registration
config vsp session-config-pool entry name registration
```

Properties

Property name	Description
admin {enabled | disabled}	Specifies whether to accept or deny REGISTER requests. If set to **enabled**, the system accepts REGISTERs; the system denies REGISTERs if set to **disabled**. **Example: set admin enabled** The default setting is **disabled**.
cache-lcs {enabled | disabled}	Specifies whether the system should add entries for LCS clients to the registration database. Use this object in single-domain configurations, where the system sits between the client and the LCS server. When **enabled**, client requests that pass through the system are added to the database. The system adds an entry for the LCS client record into its registration database. Therefore, you can enable features that use the database, such as call forking. When **disabled**, only client requests from external domains are added to the database. The system passes all call requests directly to the LCS server. **Example: set cache-lcs enabled** The default setting is **disabled**.

Property name	Description
add-path-header {enabled | disabled}	Specifies whether the system adds a Path: header to the packet. The Path header, with syntax similar to the Record-Route header, is used in conjunction with REGISTER requests and with 2xx messages. **Example: set add-path-header enabled** The default setting is **disabled**.
local-options-reply {enabled | disabled}	Specifies how the system responds to OPTIONS requests during a registration session. When **enabled**, the system flags the matching AOR entry in the location cache with this setting. In future registration sessions for that AOR, the system will respond locally to any OPTIONS requests, rather than forwarding them to the phone. When **disabled**, OPTIONS requests are forwarded normally. **Example: set local-options-reply enabled** The default setting is **disabled**.
handle-3xx-locally {enabled | disabled}	Specifies whether the system forwards responses to 3xx messages back to the UAC or resends them. If **disabled**, when the system receives a 3xx response for a REGISTER (e.g., 301 Moved Permanently or 302 Moved Temporarily), it forwards the response back to the client. When this option is **enabled**, the system does not forward the response back. Instead, it hunts through the contact routes in the response message until it finds one that responds with a 200 OK. It then applies that route to the REGISTER message. **Example: set handle-3xx-locally enabled** The default setting is **disabled**.

Property name	**Description**
delegate-aged-bindings {enabled | disabled}	Specifies whether the system can redelegate a REGISTER request before both sides experience a binding expiration. By default, when the binding expires on the client-side, the system does not redelegate the REGISTER until the peer expiration. This can cause problems if, for example, you are using DNS to load balance REGISTERs between systems. In that case, the REGISTER may not be redelegated to the third-party server until the peer timer expires, making the server unaware of the relocation. When **enabled**, the system redelegates the REGISTER request to the peer even if only the client-side timer has expired. **Example: set delegate-aged-bindings enabled** The default setting is **disabled**.
unregister-aged-bindings {enabled | disabled}	Specifies whether AA-SBC allows delegation of an unregister-all (*) request. When **enabled**, AA-SBC delegates the Contact: * UNREGISTER request unchanged and removes all bindings associated with the AOR that have this feature enabled. **Example: set unregister-aged-bindings enabled** The default setting is **disabled**.
delegate-unregister-all {enabled | disabled}	Specifies whether the system sends an UNREGISTER request when a binding ages out. When enabled, if a binding ages out the system sends an UNREGISTER to the delegate. The system also resets the peer expiration so that the next REGISTER from this client is delegated. You can remove the binding without sending the UNREGISTER by using the location **flush now** action. **Example: set delegate-unregister-all enabled** The default setting is **disabled**.

authentication

Purpose

Sets the authentication mode to use on this SIP session. For authentication services that involve remote servers (such as RADIUS and DIAMETER), you must configure these servers on AA-SBC using the either the radius-group or diameter-group configuration objects. For the directory authentication mechanism, you must first configure the directory service in the directory configuration object.

Syntax

```
config vsp default-session-config authentication
config vsp policies session-policies policy name rule name
   session-config authentication
config vsp dial-plan dial-prefix entryName session-config
   authentication
config vsp dial-plan route name session-config authentication
config vsp dial-plan source-route name session-config authentication
config vsp session-config-pool entry name authentication
```

Properties

Property name	Description
mode {none [enabled | disabled] | Local [enabled | disabled] | RADIUS [enabled | disabled] *radiusGroupReference* | DIAMETER [enabled | disabled] *diameterGroupReference* | Directory [enabled | disabled] *directoryReference*}	Sets the type of authentication the system uses for this SIP session. Optionally, you can set whether the authentication applies to inbound-only (**enabled**) or inbound and outbound (**disabled**) traffic. Select either: • **none**—the system performs no authentication. • **Local**—the system uses the username and password configured in the user object for authentication. • **RADIUS**—the system performs RADIUS authentication according to the configuration specified in the radius-group object. • **DIAMETER**—the system performs DIAMETER authentication according to the configuration specified in the diameter-group object. • **Directory**—the system expects the user credentials that are specified in the directory service that you supply. **Example: set mode radius enabled "vsp radius-group boston1"** The default setting is **none** for the authentication type and **disabled** for the inbound-only setting.
session-starter-only {enabled | disabled}	Specifies which requests the system challenges. When **disabled** (the default), if authentication is enabled the system challenges all requests in a session. When **enabled**, the system only challenges the first request in a session. **Example: set session-starter-only enabled.** The default setting is **disabled.**

Property name	Description
handle-challenge-locally {enabled | disabled}	Sets whether a challenge is handled locally. When **enabled**, the system terminates the original challenge response (either 401 Unauthorized or 407 Proxy Authentication Required) and generates a new request with the authentication information. When **disabled**, the system forwards the 401/407 response back to the UAC. **Example: set handle-challenge-locally enabled** The default setting is **disabled**.
challenge-response-code {401 | 407}	Sets the response code that the system sends when it terminates the original challenge response (either 401 Unauthorized or 407 Proxy Authentication Required). This code is only applied when the **handle-challenge-locally** property is enabled. **Example: set challenge-response-code 407** The default setting is **401** (Unauthorized).
apply-to-methods *messageTypes*	Specifies which message types to authenticate. This setting is used by the **registration-throttling** property of the route and source-route registration plan objects to define which message types require authentication. When you modify this value, the system overwrites the current setting with only the message types you specify. For example, if set to the default and you enter **INVITE**, the system only authenticates INVITE messages. Enter multiple types separated by a plus sign (+) with no spaces. **Example: set apply-to-methods INVITE+REFER+REGISTER** The default setting is **INVITE+REGISTER+BYE**.
exclude-scheme-in-called-station {false | true}	*Secondary property.* Specifies which portions of the TO URI that AA-SBC uses for authentication. If set to **false**, the system authenticates the full TO URI. If **true**, the system uses only the User and Host portions of the TO URI. **Example: set challenge-response-code 407** The default setting is **false**.

authorization

Purpose

Sets the type of authorization the system performs for matching sessions. Using this object, you set the protocol the system uses to get authorization data— either none, the least-cost routing engine, or WSDL. This data results in a list of routing options for the call. For LCR, you must configure the route-server master service and, for intercluster lookups, a diameter client and server. For WSDL, you configure an external-services policy-group.

Syntax

```
config vsp default-session-config authorization
config vsp policies session-policies policy name rule name
   session-config authorization
config vsp dial-plan dial-prefix entryName session-config
   authorization
config vsp dial-plan route name session-config authorization
config vsp dial-plan source-route name session-config authorization
config vsp session-config-pool entry name authorization
```

Properties

Property name	Description
mode {None | Local | WSDL *policyGrpReference* [true | false] [true | false] | Diameter *diameterGrpReference*}	Sets the method to use for authorization data retrieval. Select either: • **None**—the system performs no LCR lookup. This is the equivalent of administratively disabling the LCR service for matching calls. • **Local**—the system performs intracluster LCR lookup; it sends the route lookup request to the system hosting the route-server master service. • **WSDL**—the system sends the request to the external services policy server specified in the policy-group configuration. Optionally, you can specify whether to send SIP message headers and/or content (both default to **false**) with the request. • **Diameter**—the system sends the route request to the server specified in the diameter-group object configuration. This is typically only used for intercluster lookup configurations. • RADIUS—the AA-SBC sends a request to the RADIUS server with the to-URL and from-URL in the request. The RADIUS server responds with information that the AA-SBC uses to create session-configs that are applied to the session. **Example: set mode Diameter “vsp diameter-group LCRserver”** The default setting is **None**.

Property name	Description
always-perform-lookup {true | false}	Specifies whether the system should do an authorization lookup (if configured to do so with the **mode** property). If set to **true**, the default, the system retrieves authorization data regardless of other configuration settings. If set to **false**, the system first uses internal logic to determine whether session handling data can be derived from other sources (e.g., location cache or dial plan). Set this to **false**, for example, when handling two locally registered phones calling each other. **Example: set always-perform-lookup false** The default setting is **true**.
apply-to-methods *messageTypes*	Specifies to which message types the system applies authorization processing. When you modify this value, the system overwrites the current setting with only the message types you specify. For example, if set to the default and you enter **INVITE**, the system only authorizes INVITE messages. Enter multiple types separated by a plus sign (+) with no spaces. **Example: set apply-to-methods INVITE+REGISTER** The default setting is **INVITE**.

accounting

Purpose

Sets the target destination for call detail records. By selecting a target, you are configuring AA-SBC to provide call logging of this SIP session. The records are then sent to the server, database, or file specified in this object. Note that you must configure the destination devices first, and then reference them here.

Syntax

```
config vsp default-session-config accounting
config vsp policies session-policies policy name rule name
   session-config accounting
```

```
config vsp dial-plan dial-prefix entryName session-config accounting
config vsp dial-plan route name session-config accounting
config vsp dial-plan source-route name session-config accounting
config vsp session-config-pool entry name accounting
```

Properties

Property name	Description
target {radius *radiusGroup* | diameter *diameterGroup* | database *databaseGroup* | syslog *syslogGroup* | file-system *path*}	Sets the destination for the accounting records (SIP call detail records) created for this SIP session. When you set the target, you must specify a previously configured object, dependent on the target type: • **radius**—logs the session to the group specified in the radius-group object. • **diameter**—logs the session to the group specified in the diameter-group object. • **database**—logs the session to an internal or external database group, as specified in the database object diameter-group subobject. • **syslog**—logs the session to an external syslog server group, as specified in the syslog object diameter-group subobject. • **file-system**—writes the session to a file on the AA-SBC device, as specified in the file-system object path subobject. **Example: set target syslog “vsp accounting syslog group Boston”** There is no default setting.
header *string*	Specifies a string that is written to each accounting record. Use this, for example, to track for later analysis a header that certain user agents output in their INVITE. This header can be seen in the **Arbitrary Header** field of the Call Record displayed through the AA-SBC Management System **Call Logs** tab. **Example: set header UA1** There is no default setting.
accept-mode {enabled | disabled}	Specifies the activity that initiates the connect time in the accounting record. When **disabled**, the default, the connect time is recorded when the system transmits an ACK. When **enabled**, the connect time is recorded when the system receives a 200 OK message. **Example: set accept-mode enabled** The default setting is **disabled**.

Property name	Description
disconnect-time-upon-receipt-of-bye {enabled | disabled}	Enables or disables logging of the call disconnect time entry in the call detail record when the call session terminates with a BYE request. If set to **disabled** (the default), the disconnect time is recorded with the 200OK that follows the BYE request. **Example: set disconnect-time-on-receipt-of-bye enabled** The default setting is **disabled**.
use-short-gateway-names {enabled | disabled}	Specifies how AA-SBC handles particular gateway fields in the CDR. When **enabled** (the default), the system populates the OrigGW and TermGW fields in the CDR with the server-pool server-pool-admission-control name string for the originating and terminating SIP server gateway for a call, if known. If set to **disabled**, AA-SBC populates the OrigGW and TermGW fields in the CDR with the server name (as configured) followed by a "- "and then the server-pool server-pool-admission-control name. **Example: set use-short-gateway-names disabled** The default setting is **enabled**.
report-failed-calls {enabled | disabled}	*Secondary property*. Specifies whether to send out accounting records for calls that did not connect. When **disabled**, the records are not sent. When **enabled**, a record of the call is sent to the target configured in your session configuration. **Example: set report-failed-calls enabled** The default setting is **disabled**.

log-alert

Purpose

Enables or disables session logging on a per-call basis. Because session logging can result in large amounts of data, this object allows you to enable logging only on calls that match certain criteria. Note that logging must also be enabled in the event-log object for the session details to be written to a target.

Syntax

```
config vsp default-session-config log-alert
config vsp policies session-policies policy name rule name
   session-config log-alert
config vsp dial-plan dial-prefix entryName session-config log-alert
config vsp dial-plan route name session-config log-alert
config vsp dial-plan source-route name session-config log-alert
config vsp session-config-pool entry name log-alert
```

Properties

Property name	Description
message-logging {enabled | disabled | no-registers | invite-session-only | filtered}	Enables or disables logging of individual SIP messages (INVITE, CANCEL, BYE, etc.) to the system database. Select **no-registers** to log all but SIP REGISTER messages. (This can also be accomplished with the master-services database object **sip-register** property, but message-logging is the preferred method.) If you select **enabled**, use the **apply-to-methods-for-filtered-logs** property to set the message type. Message logging can only be enabled if you have the appropriate license for the feature. The system uses the database records to: • Display the call detail diagrams at the AA-SBC Management System and logs which show the individual SIP messages involved in each session. • Provide collected messages for detection of attack patterns, which are then used to construct DOS rules blocking the attack traffic. **Example: set message-logging enabled** The default setting is **enabled** if you have the license for message logging features and **disabled** if you do not.
apply-to-methods-for-filtered-logs *msgType*	Specifies the type of SIP message to be logged if **message-logging** is set to enabled. If set to any other option, this property is ignored. **Example: set apply-to-methods-for-filtered-logs INVITE+REFER** The default setting is **INVITE**.

Property name	Description
alert {enabled "*logTargetReference*" severity | disabled}	*Secondary property.* Enables or disables the sending of session alert messages to the configured system logging target. If set to **enabled**, specify the path to a previously configured logging target and a severity level. The path must be specified in quotation marks. The severity levels are: • emerg(ency) • alert • crit(ical) • error • warning • notice • info(rmation) • debug **Example: set alert enabled "services event-log file messages" info** The default setting is **disabled**.
logging {enabled | disabled}	*Secondary property.* Enables or disables event logging for this session. If session logging is **enabled**, and the event-log object enables logging, details are recorded in a target file.If session logging is **disabled**, even if the event-log is enabled, the system does not write session events to the log. **Example: set logging enabled** The default setting is **disabled**.
tracing {enabled | disabled}	*Secondary property.* Enables SIP-related tracing for the session. When **enabled**, you can exit to a SIP shell and enter the **trace-filter** command to see related traces. (Note that you must have advanced CLI permissions to execute shell commands.) **Example: set tracing enabled** The default setting is **disabled**.
`message-auditing {enabled severity | disabled}`	*Secondary property.* Enables the system to maintain an audit trail of changes to each SIP message. **Example: set message-auditing enabled** The default setting is **disabled**. If enabled, the default severity level is **error**.

refer-settings

Purpose

Enables or disables call parking compatibility settings for the Sylantro SIP for Business initiative.

Syntax

```
config vsp default-session-config refer-settings
config vsp policies session-policies policy name rule name
   session-config refer-settings
config vsp dial-plan dial-prefix entryName session-config
   refer-settings
config vsp dial-plan route name session-config refer-settings
config vsp dial-plan source-route name session-config refer-settings
config vsp session-config-pool entry name refer-settings
```

Properties

Property name	Description
modify-call-parking {enabled | disabled}	Enables or disables call parking compatibility features. When **enabled**, the system modifies the Refer-To header. Leave this property at the default setting of **disabled** if you are not using the SIP for Business platform. **Example: set modify-call-parking enabled** The default setting is **disabled**.
fix-refer-to-call-id {enabled | disabled}	Specifies whether to modify the call ID value in the Refer-to header. Enable this property if there are problems, caused by an incorrectly implemented NAT device, with the Replaces section of the Refer-to header. When **enabled**, if the system detects a *@ipAddress* string at the end of the Refer-to header call-id field, the system replaces the address with the message's remote IP address. This allows the system to later use that value to find the call leg it refers to and perform a correct translation. **Example: set fix-refer-to-call-id enabled** The default setting is **disabled**.

group-settings

Purpose

Creates a tag (group name) that can be stored in the location cache during registration. When AA-SBC receives a REGISTER and applies a matching session configuration, it saves out the zero or more groups (configured with this object) that are associated with that session config. When AA-SBC receives an INVITE destined for a registered phone, a location cache lookup results in return of all the stored group names, which can then be used to further refine the selection of the applicable session config for the INVITE. You can use this feature, for example, to control outbound settings that are specific to a type of phone, such as the encryption type or CODEC preferences.

Syntax

```
config vsp default-session-config group-settings
config vsp policies session-policies policy name rule name
   session-config group-settings
config vsp dial-plan dial-prefix entryName session-config
   group-settings
config vsp dial-plan route name session-config group-settings
config vsp dial-plan source-route name session-config group-settings
config vsp session-config-pool entry name group-settings
```

Properties

Property name	Description
group-name *string*	Specifies the name of the group. The system transfers each group name in a matching session config to the location cache to be stored with the registration. **Example: set group-name eyebeam_group** There is no default setting.

instant-messaging

Purpose

Enables IM archiving, applies text stamps before or after IM messages, and sends IM message alerts to the configured AA-SBC event log.

Syntax

```
config vsp default-session-config instant-messaging
config vsp policies session-policies policy name rule name
   session-config instant-messaging
config vsp dial-plan dial-prefix entryName session-config
   instant-messaging
config vsp dial-plan route name session-config instant-messaging
config vsp dial-plan source-route name session-config
   instant-messaging
config vsp session-config-pool entry name instant-messaging
```

Properties

Property name	Description
directive {allow | discard | refuse [*resultCode*] [*resultString*] | follow-sip-directive}	Assigns an action to the message. Select one of the following actions: • **allow**—allows the message, even if higher-level policy (under instant-messaging) says to refuse or discard it. • **discard**—silently deletes the message instead of delivering it. No notification is sent. • **refuse**—deletes the message, but sends a SIP error response to the sending agent. Optionally, specify the result code, between 400 and 699, and/or a result string to send in the error response. The default error code is 400, with no accompanying text. • **follow-sip-directive**—follows whatever actions are configured at the session-level. (These are the settings under sip-directive, and/or instant-messaging.) Note that if you specify the **refuse** directive with text, the text is placed on the method line of the SIP response message. That line is usually not displayed to the user. If you want a message displayed to the sender, use the **message-to-sender** property. **Example: set directive refuse 500 “Message discarded”** The default setting is **follow-sip-directive**.

Property name	Description
alert {enabled | disabled} “logTargetPath” severity	Sends alert messages containing IM session information and message content to the configured system event log. Specify the configured event logging target path and severity level. Severity levels: • emerg(ency) • alert • crit(ical) • error • warning • notice • info(rmation) • debug **Example: set alert enabled “services event-log file messages” info** The default setting is **disabled**
archiving {enabled | disabled}	Enables or disables archiving of SIP instant messages to the system database. When **enabled**, the system records to its database all instant messages (the text in the body of MESSAGE SIP messages) of sessions matching the policy. Messages are recorded in both directions. Note that you must enable the **database-write** property of the vsp object for archiving to work. You can view the instant messages that have been archived using the AA-SBC Management System **Call Logs** feature. **Example: set archiving enabled** The default setting is **disabled**
pre-stamp “*text*”	Prepends the user-specified text before the IM message content in this SIP session. **Example: set pre-stamp “Good Morning:”** There is no default setting.

Property name	Description
post-stamp "*text*"	Appends the specified text after the IM message content in this SIP session. **Example: set post-stamp "Have a great day"** There is no default setting.
message-to-sender "*text*"	Sets the text message to send back to the originating IM sender in this SIP session. **Example: set message-to-sender "Messages to this user are logged."** There is no default setting.
message-to-recipient "*text*"	Sets the text message to send to the IM recipient in this SIP session. This text is in addition to the incoming message. **Example: set message-to-recipient "Message is being logged."** There is no default setting.

instant-messaging-content

Purpose

Creates pointers to configured word lists and/or URL lists. For more detailed information on instant message filtering, see Chapter 34, "IM Filtering objects". To create word lists, see the word-list object; to create URL lists, see the url-list object.

Syntax

```
config vsp default-session-config instant-messaging-content
config vsp policies session-policies policy name rule name
   session-config instant-messaging-content
config vsp dial-plan dial-prefix entryName session-config
   instant-messaging-content
config vsp dial-plan route name session-config
   instant-messaging-content
config vsp dial-plan source-route name session-config
   instant-messaging-content
config vsp session-config-pool entry name instant-messaging-content
```

Properties

Property name	Description
word-list *reference*	Configures a pointer to a previously configured word list. You can include any number of words lists in your instant messaging content scan. **Example: set word-list "vsp im-filtering word-list bad-words"** There is no default setting
url-list *reference*	Configures a pointer to a previously configured URL list. You can include any number of URL lists in your instant messaging content scan. **Example: set url-list "vsp im-filtering url-list good-guys"** There is no default setting.

file-transfer

Purpose

Enables recording of file transfers on AA-SBC.

Syntax

```
config vsp default-session-config file-transfer
config vsp policies session-policies policy name rule name
   session-config file-transfer
config vsp dial-plan dial-prefix entryName session-config
   file-transfer
config vsp dial-plan route name session-config file-transfer
config vsp dial-plan source-route name session-config file-transfer
config vsp session-config-pool entry name file-transfer
```

Properties

Property name	Description
anchor {enabled | disabled}	Enables or disables anchoring, which defines whether the system is used as an intermediary for traffic. When **enabled**, all file transfers pass through the AA-SBC device. If **disabled**, transfers circumvent the AA-SBC device. (The system would still have a record of the transfer, however, because it keeps records of all SIP transactions.) You must enable anchoring to use the recording or virus scanning features. **Example: set anchor enabled** The default setting is **disabled**.
record {enabled | disabled}	Enables or disables recording of file transfers on the system in this SIP call session. The file is stored on the system and forwarded to the SIP call recipient. Anchoring must be enabled to use the recording feature. **Example: set record enabled** The default setting is **disabled**
max-filesize *bytes*	Configures the maximum file size, in bytes, allowed in a transfer through the AA-SBC device. If the size limit is exceeded, the file is dropped. **Example: set max-filesize 500000000** Enter a value between 1 and 1,073,741,824. The default setting is **1,073,741,824** bytes.

Property name	Description
allow-non-default-ports {enabled | disabled}	Specifies whether or not the system is limited in choice of ports when anchoring a file transfer. When **enabled**, the default, the system can use any port. Leave this setting for interoperability with LCS 2005. When **disabled**, the system can only use the default port that is specified by Microsoft. **Example: set allow-non-default-ports disabled** The default setting is **enabled**.
t120-anchor {enabled | disabled}	Enables anchoring of the application sharing and whiteboard features found in the Office Communicator 2005 client (based on ITU T.120). If you **enable** this feature, you must also enable the anchor property of the media object. If you are anchoring federated traffic, you must also set the sip-settings **lcs-compatibility** bit to compensate for a bug in the OC 2005 client that does not accept Content-Type headers with the subtype “SDP” specified in uppercase. Set the bit to 0x010032e3. **Example: set t120-anchor enabled** The default setting is **disabled**.

forking-settings

Purpose

Configures the ring pattern for SIP calls from AA-SBC to the endpoint. AA-SBC selects ring destinations based on information in the address of record (AOR) from the locations services database.

You can also configure an ordered set of rules (metrics) to influence the final server (or next hop) selection for outbound calls with the **outbound-arbiter-rule** property. This may be necessary if you have configured multiple possible next-hop devices using the peer object. The arbitration calculation rules apply to all outbound devices to determine the destination. This set of rules takes precedence over any arbitration decision made through the dial-plan arbiter configuration. If a dial-plan references a session-config in which this object **outbound-arbiter-rule** is configured, AA-SBC ignores the dial-plan arbiter configuration and uses this one instead.

Syntax

```
config vsp default-session-config forking-settings
config vsp policies session-policies policy name rule name
   session-config forking-settings
config vsp dial-plan dial-prefix entryName session-config
   forking-settings
config vsp dial-plan route name session-config forking-settings
config vsp dial-plan source-route name session-config forking-settings
config vsp session-config-pool entry name forking-settings
```

Properties

Property name	Description
forking-type {none | sequential | parallel}	Sets the AA-SBC forwarding behavior when it receives a call INVITE. This is the method in which the system forwards the call to the endpoint. In order for this property to work properly, the **outbound-arbiter-rule** parameter must be configured. • **none**—the system forwards the call to the latest binding it finds for the Request URI in the location service. • **sequential**— the call is forwarded to each binding for each AOR stored for a destination, one at a time, until it is successful. If the system receives a fail message, busy, or timeout, it tries the next AOR. The delay between trying each device is set with the **session-provisional-timeout** property of the sip-settings object. • **parallel**—the call is forwarded to each binding for all AORs for the destination. **Example: set forking-type sequential** The default setting is **none**.
max-hunt *integer*	Specifies the number of destinations to try when the system is configured to do sequential forking. Setting this value prevents the creation of a forking loop in the event that a server redirects a call to another server in the listed destinations. **Example: set max-hunt 50** The default setting is **100** destinations.

Property name	Description
outbound-arbiter-rule *algorithm*	Enters rules into the arbiter configuration. Enter as many rules as you wish. If you do not set any rules, the system uses the settings of the dial-plan arbiter (or factory defaults if the dial-plan also has no arbiter configuration). If you select least-cost, you can optionally set a maximum (or unlimited) value for call cost. It you select trunk-qos, you can optionally select a previously configured class-of-service. This property is required for the **forking-settings** properties to work properly. See the Routing algorithm options table for a description of the selections. **Example: set rule least-cost 15** There is no default setting.
max-arbitration-options	*Secondary property*. Specifies the number of potential destinations to consider when applying a rule. The smaller of this and **max-hunt** takes effect when the final destinations are determined. **Example: set max-arbitration-options 50000** Min: 0 / Max: 7294967295 The default setting is **unlimited**.

header-settings

Purpose

Configures AA-SBC to remove, or to remove and replace the content of fields from the SIP header. In addition, you can identify header types to specifically allow or block. The allowed-headers and blocked-headers properties apply the following rules:

- The From, To, CSeq, and Call-ID headers are required and cannot be blocked.
- The allowed list overrides settings of the blocked list. If a header is explicitly allowed, it cannot then be blocked using the blocked list.
- If a header name does not match a value in either list, it is allowed.
- AA-SBC accepts regular expressions for an entry; all special characters apply.

Syntax

```
config vsp default-session-config header-settings
config vsp policies session-policies policy name rule name
   session-config header-settings
config vsp dial-plan dial-prefix entryName session-config
   header-settings
config vsp dial-plan route name session-config header-settings
config vsp dial-plan source-route name session-config header-settings
config vsp session-config-pool entry name header-settings
```

Properties

Property name	Description
allowed-header *headerString*	Sets the SIP headers that should be explicitly allowed to remain in the SIP message. You can enter any number of header names by re-executing the command. See the Purpose for applicable rules. **Example: set allowed-header Via** There is no default setting.
blocked-header *headerString*	Sets the SIP headers that should be explicitly removed from the SIP message. You can enter any number of header names by re-executing the command. See the Purpose for applicable rules. **Example: set blocked-header .*** There is no default setting.
apply-allow-block-to {requests | responses | requests-and-responses}	Sets whether the allow and block properties of this object apply to requests only or requests and responses. When **disabled**, changes apply only to requests. When **enabled**, the default, changes apply to requests and responses. **Example: set apply-allow-block-to responses** The default setting is **requests-and-responses.**.

Property name	Description
pAssert-mode {disabled | enabled}	*Secondary property.* Sets whether to strip the number in the P-Asserted-Identity field from the SIP header. When **enabled**, the system replaces the value in the From field with the value from the P-Asserted-Identity field for the outbound call leg. (Note that the system maintains the original From field value in the Contact field.) **Example: set mode enabled** The default setting is **disabled**.
header-to-strip *fieldName*	*Secondary property.* Configures the system to strip the value of the specified field. Enter a SIP header field name. **Example: set header-to-strip Remote-Party-ID** There is no default setting.

altered-header

Purpose

Modifies or creates header values in calls matching this session configuration. Both this and the reg-ex-header objects provide this functionality. Use this object, which is simpler, when possible. Use the **reg-ex-header** object for complex modification, for example, when multiple levels of change are required. Note that you can create multiple header-altering configurations. They are processed by the system in the order that they appear in the configuration.

Syntax

```
config vsp default-session-config header-settings altered-header
   number
config vsp policies session-policies policy name rule name
   session-config header-settings altered-header number
config vsp dial-plan dial-prefix entryName session-config
   header-settings altered-header number
config vsp dial-plan route name session-config header-settings
   altered-header number
config vsp dial-plan source-route name session-config header-settings
   altered-header number
config vsp session-config-pool entry name header-settings
   altered-header number
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables this configuration entry. **Example: set admin enabled** There is no default setting.
source-header *headerString*	Specifies the URI from which the system initially derives the data that is to be written to the destination header. **Example: set source-header to** There is no default setting.
source-field {user | host | selection *regEx replacement* | value replacement}	Specifies the portion of the URI that the system writes to the destination. Select either **user**, **host**, **selection**, or **value**. With the **user** and **host** options, the system writes the entire field to the destination. If you choose **selection**, specify the value within the URI to match on and the replacement text to write to the destination. Or, select **value** to write a specific string to the destination. In this case, the **source-header** field is ignored. **Example: set source-field user** There is no default setting. For more information regarding configuring regular expressions and replacement strings, see the section in Chapter 1, “Using regular expressions” on page 36.
destination-header *headerString*	Specifies the header to be created or modified by the properties set in this object. The system modifies this URI with the data from the source. If the header does not exist in the message, the system creates it. **Example: set destination-header request** There is no default setting.
destination-field {user | host | display | full}	Specifies the field in the specified destination URI to overwrite. Select either **user**, **host** , or **display**. To overwrite the entire selected destination URI, select **full**. **Example: set destination-field user** There is no default setting.

Property name	Description
apply-to-methods *messageTypes*	Specifies the message type to which the system applies header value changes. The system then changes the specified URI according to the settings of the header and destination properties of this object. When you modify this value, the system overwrites the current setting with only the message types you specify. For example, if set to the default and you enter **INVITE**, the system only authenticates INVITE messages. Enter multiple types separated by a plus sign (+) with no spaces. **Example: set apply-to-methods INVITE+REFER** The default setting is **INVITE**.
apply-to-responses {no | yes *responseCode*}	Specifies whether to apply header value changes to SIP requests or requests and responses. Set to **no** to apply changes only to requests. Set to **yes** to apply to responses as well. If yes, you must set the response code to which it applies. Create additional altered-body profiles to change multiple response types. **Example: set apply-to-responses yes 200** The default setting is **no**.

Property name	Description
session-persistent {enabled | disabled}	Specifies to which messages in a session SessionManager should apply changes made with this object. When **enabled**, AA-SBC applies any TO, FROM, or REQUEST URI changes to the first and all subsequent messages in a session. When **disabled**, the default, the system applies the changes only to the first message in the session. **Example: set session-persistent enabled** The default setting is **disabled.**.
cseq *integer*	*Secondary property.* Sets a mechanism to further filter which SIP messages have the header expression modifications applied. If **cseq** is set to zero (the default), AA-SBC applies the changes to all SIP messages. If set to any other value, the system only applies the changes to SIP messages having a CSEQ field that matches that value. **Example: set cseq 100** The default setting is **0.**.

reg-ex-header

Purpose

Provides granular modification capabilities to create or modify a header. Use this when the methods of the altered-header object do not achieve the necessary changes. You can create multiple reg-ex-header entries to accomplish complicated manipulations. The system executes each one, the order of execution based on their order within the configuration.

Syntax

```
config vsp default-session-config header-settings reg-ex-header number
config vsp policies session-policies policy name rule name
   session-config header-settings reg-ex-header number
config vsp dial-plan dial-prefix entryName session-config
   header-settings reg-ex-header number
config vsp dial-plan route name session-config header-settings
   reg-ex-header number
config vsp dial-plan source-route name session-config header-settings
   reg-ex-header number
config vsp session-config-pool entry name header-settings
   reg-ex-header number
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables this configuration entry **Example: set admin enabled** The default setting is **enabled**.
destination *headerString*	Specifies the header to be created or modified by the properties set in this object (the header to write the changes to). If the header doesn't exist in the message, the system creates it. When you configure this, you must also set either the create or append properties. Otherwise, there is no configuration from which to derive the new URI. You do not specify the fields within the destination to change as that level of change is accomplished with the regular expression statements. To modify an existing header, use the same header for the both the **destination** and the **create** properties. **Example: set destination Diversion** There is no default setting.
create *source expression replacement*	Identifies the header containing the data to be modified and then written to the destination header. (Use the **append** property to add data to the existing header.) First, select the header that serves as the source of the data. Then, specify a regular expression to run against the value of the source header. Also supply the replacement expression to apply if there's a match. Changes are only applied to the original expression match. **Example: set create History-Info "([^,<]*?)<sip:([^>?]*?)\?Reason=SIP%3Bcause%3D3[0-9]{2}(.*?)>(.*?)index=(\d.\d.\d|\d.\d|\d)" "<sip:\2>;reason=deflection"** There is no default setting. For more information regarding configuring regular expressions and replacement strings, see the section in Chapter 1, “Using regular expressions” on page 36.

Property name	Description
append *source expression replacement*	Identifies the header containing the data to be added to the destination header. First, select the header that serves as the source of the data. Then, specify a regular expression to run against the value of the source header. Also supply the replacement expression to apply if there's a match. The system appends this string to the existing destination header. To add spaces or commas, be sure to include them (using quotation marks) in the replacement statement. **Example: set append History-Info ([^,<]*?)<sip:cov.com ", <sip:cov2.com"** The default setting is **enabled**. For more information regarding configuring regular expressions and replacement strings, see the section in Chapter 1, “Using regular expressions” on page 36.
apply-to-methods *messageTypes*	Specifies the message type to which the system applies header value changes. The system then changes the specified URI according to the settings of the header and destination properties of this object. When you modify this value, the system overwrites the current setting with only the message types you specify. For example, if set to the default and you enter **INVITE**, the system only authenticates INVITE messages. Enter multiple types separated by a plus sign (+) with no spaces. **Example: set apply-to-methods INVITE+REFER** The default setting is **INVITE**.

Property name	Description
apply-to-responses {no | yes *responseCode*}	Specifies whether to apply header value changes to SIP requests or requests and responses. Set to **no** to apply changes only to requests. Set to **yes** to apply to responses as well. If yes, you must set the response code to which it applies. Create additional altered-body profiles to change multiple response types. **Example: set apply-to-responses yes 200** The default setting is **no**.
session-persistent {disabled | enabled}	Specifies to which messages in a session AA-SBC should apply changes made with this object. When **enabled**, AA-SBC applies any TO, FROM, or REQUEST URI changes to the first and all subsequent messages in a session. When **disabled**, the default, the system applies the changes only to the first message in the session. **Example: set session-persistent enabled** The default setting is **disabled**.
cseq *integer*	*Secondary property.* Sets a mechanism to further filter which SIP messages have the header expression modifications applied. If **cseq** is set to zero (the default), AA-SBC applies the changes to all SIP messages. If set to any other value, the system only applies the changes to SIP messages having a CSEQ field that matches that value. **Example: set cseq 100** The default setting is **0.**.

header-normalization

Purpose

Modifies the User portion of the specified header. This object uses the same methodology as the dial-plan normalization object.

Syntax

```
config vsp default-session-config header-settings header-normalization
   number
config vsp policies session-policies policy name rule name
   session-config header-settings header-normalization number
config vsp dial-plan dial-prefix entryName session-config
   header-settings header-normalization number
config vsp dial-plan route name session-config header-settings
   header-normalization number
config vsp dial-plan source-route name session-config header-settings
   header-normalization number
config vsp session-config-pool entry name header-settings
   header-normalization number
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables this configuration entry **Example: set admin enabled** The default setting is **enabled**.
destination *headerString*	Specifies the header to be normalized. The system makes changes to the User field of the destination URI. Changes are applied to all messages types identified in the **apply-to-methods** property. **Example: set destination Diversion** There is no default setting.

Property name	Description
value *setting*	Sets the type of normalization that the system applies to outgoing calls to a provider (to the USER field of the destination URI). See User normalization properties for property setting options and descriptions. **Example: set value replace-prefix 866** The default type setting is **none** (no normalization applied).
apply-to-methods *messageTypes*	Specifies the message type to which the system applies header value changes. The system then changes the specified URI according to the settings of the **value** property of this object. When you modify this value, the system overwrites the current setting with only the message types you specify. For example, if set to the default and you enter **NOTIFY**, the system only modifies NOTIFY messages. Enter multiple types separated by a plus sign (+) with no spaces. **Example: set apply-to-methods INVITE+CONTACT** The default setting is **INVITE**.
apply-to-responses {no | yes *responseCode*}	Specifies whether to apply header value changes to SIP requests or requests and responses. Set to **no** to apply changes only to requests. Set to **yes** to apply to responses as well. If yes, you must set the response code to which it applies. Create additional altered-body profiles to change multiple response types. **Example: set apply-to-responses yes 200** The default setting is **no**.

Property name	Description
session-persistent {disabled | enabled}	Specifies to which messages in a session AA-SBC should apply changes made with this object. When **enabled**, AA-SBC applies any TO, FROM, or REQUEST URI changes to the first and all subsequent messages in a session. When **disabled**, the default, the system applies the changes only to the first message in the session. **Example: set session-persistent enabled** The default setting is **disabled**.
cseq *integer*	*Secondary property.* Sets a mechanism to further filter which SIP messages have the header expression modifications applied. If **cseq** is set to zero (the default), AA-SBC applies the changes to all SIP messages. If set to any other value, the system only applies the changes to SIP messages having a CSEQ field that matches that value. **Example: set cseq 100** The default setting is **0.**.

altered-body

Purpose

Modifies the SIP message body in calls matching this session configuration. AA-SBC searches the SIP message body for the specified string and replaces the string as required.

Syntax

```
config vsp default-session-config header-settings altered-body number
config vsp policies session-policies policy name rule name
   session-config header-settings altered-body number
config vsp dial-plan dial-prefix entryName session-config
   header-settings altered-body number
config vsp dial-plan route name session-config header-settings
   altered-body number
config vsp dial-plan source-route name session-config header-settings
   altered-body number
config vsp session-config-pool entry name header-settings altered-body
   number
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables this configuration entry. **Example: set admin enabled** There is no default setting.
altered-body *regEx replacement*	Specifies the text to match on in the message body and the replacement text for the matched string. **Example: set altered-body "(?ms)(.*)<address uri=\""sip:(.*)@(.*):5060;(.*)" "\1<address uri=" "sip:\2@10.1.1.1:5060;\4"** There is no default setting. For more information regarding configuring regular expressions and replacement strings, see the section in Chapter 1, “Using regular expressions” on page 36.

Property name	Description
apply-to-methods *messageTypes*	Specifies the message type to which the system applies message body changes. The system then changes the SIP message body (if a match occurs) in all messages of that type. When you modify this value, the system overwrites the current setting with only the message types you specify. For example, if set to the default and you enter **CONTACT**, the system only alters CONTACT messages. Enter multiple types separated by a plus sign (+) with no spaces. **Example: set apply-to-methods INVITE+CONTACT** The default setting is **INVITE**.
apply-to-responses {no | yes *responseCode*}	Specifies whether to apply message body changes to SIP requests or requests and responses. Set to **no** to apply changes only to requests. Set to **yes** to apply to responses as well. If yes, you must set the response code to which it applies. Create additional altered-body profiles to change multiple response types. **Example: set apply-to-responses yes 200** The default setting is **no**.

Property name	Description
cseq *integer*	*Secondary property.* Sets a mechanism to further filter which SIP messages have the header expression modifications applied. If **cseq** is set to zero (the default), AA-SBC applies the changes to all SIP messages. If set to any other value, the system only applies the changes to SIP messages having a CSEQ field that matches that value. **Example: set cseq 100** The default setting is **0.**.
remove-body {true | false}	*Secondary property.* When this property is set to true, the AA-SBC removes the SIP message body from the matching of SIP messages. This includes the “Content-Type” and other related headers. Example: set remove-body true The default setting is **false**.

trusted-interface-settings

Purpose

Sets an interface to allow non-LCS devices to interact with LCS clients. The interface is a reference to a previously configured enterprise server—the LCS server that AA-SBC uses as a gateway for non-LCS traffic. The server should be configured to treat AA-SBC as authenticated.

For example, this interface would allow a SNOM phone to call an LCS Windows Messenger client. When an INVITE comes in from the SNOM phone, if the policy has the **trusted-server** property configured, AA-SBC forwards the INVITE to that server. Since the server has been configured to "treat as authenticated" traffic received from AA-SBC on that interface, it does not prompt the SNOM phone, for authentication. The INVITE is forwarded to the Windows Messenger client, and the client sees the incoming call.

Syntax

```
config vsp default-session-config trusted-interface-settings
config vsp policies session-policies policy name rule name
   session-config trusted-interface-settings
config vsp dial-plan dial-prefix entryName session-config
   trusted-interface-settings
config vsp dial-plan route name session-config
   trusted-interface-settings
config vsp dial-plan source-route name session-config
   trusted-interface-settings
config vsp session-config-pool entry name trusted-interface-settings
```

Properties

Property name	Description
trusted-server *serverReference*	Specifies the server that is configured to recognize traffic from the system as authenticated. Enter a reference to a previously configured server. **Example: set trusted-server vsp enterprise servers lcs lcs-server** There is no default setting.

session-control-settings

Purpose

Sets whether or not the system re-evaluates policy for new requests on an existing session. A session, to AA-SBC, is a SIP session between two parties, as defined by the To: and From: headers of the SIP messages.

A session is always started with a request message, such as an INVITE, SUBSCRIBE, NOTIFY, or REGISTER. Once a session is established, additional messages can be sent. For example, a request message (NOTIFY) or a MESSAGE message (containing an instant message text item) can come across an INVITE-initiated session.

Normally, AA-SBC only processes policy (or runs all of the relevant policy rules by the message) for the first message of a session. With this setting enabled, AA-SBC processes policy for each request message that comes across in a session.

Syntax

```
config vsp default-session-config session-control-settings
config vsp policies session-policies policy name rule name
   session-config session-control-settings
config vsp dial-plan dial-prefix entryName session-config
   session-control-settings
config vsp dial-plan route name session-config
   session-control-settings
config vsp dial-plan source-route name session-config
   session-control-settings
config vsp session-config-pool entry name session-control-settings
```

Properties

Property name	Description
re-evaluate-new-requests {enabled | disabled}	Specifies whether the system should process policy for all or only the first message of a session. When **enabled**, the system processes policy for each request message. When **disabled**, it processes only the first message of a session. **Example: set re-evaluate-new-requests enabled** The default setting is **disabled**.
refused-methods *messageType code text*	Provides a method to refuse the specified SIP message type. Enter the message type with a response code and accompanying error text. **Example: set refused-methods REFER 499 “REFER not allowed”** The default setting is **disabled**.
use-lnp-for-routing {enabled | disabled}	Provides a customer-specific application implementation and is not otherwise applicable. The default setting is **enabled**.
digest-realm-to-lnp-map *digestRealm lnp*	Provides a customer-specific application implementation and is not otherwise applicable. There is no default setting.
source-lnp *string*	Provides a customer-specific application implementation and is not otherwise applicable. There is no default setting.

Property name	Description
`destination-lnp` *`string`*	Provides a customer-specific application implementation and is not otherwise applicable. There is no default setting.
call-action-3pcc-server-entry *3pccServerReference*	Sets the server that will be used when a call-control action is invoked. When a server is set, AA-SBC processes any call control action on that server. If no server is specified, AA-SBC determines which server to use. **Example: set call-action-3pcc-server-entry “vsp enterprise 3pcc-servers internal-csta-server CSTA1”** There is no default setting.

playback-call-settings

Purpose

Enables or disables playback of the last recorded SIP call from a specific To/ From pair. Playback can be initiated using the dial-prefix object or any other type of policy that would trigger this object through the session configuration.

➡ **Note:** To use this feature, you must enable the anchor and record properties in the media object under the default session configuration.

For example, you could configure the **dial-prefix** to recognize *73. When a call came in with that prefix in the To or From URI of the SIP header, it would trigger the session configuration associated with that dial prefix plan. With this object **enabled**, the *73 would initiate playback of the last call. Therefore,

To:*73bob@phone.com

From:joe@phone.com

results in AA-SBC playing back, to joe@phone.com, the last call exchange that was from joe@phone.com and to bob@phone.com. AA-SBC searches the database for this call and initiates a call back to SIP phone joe@phone.com, playing back the recording of the previous call instead of connecting a new one.

You can also configure AA-SBC to play back the last call recorded call from a phone, regardless of the destination. For joe@phone.com to hear his last recorded call, he would initiate a call to himself using the configured **dial-prefix** (*73joe@phone.com).

Syntax

```
config vsp default-session-config playback-call-settings
config vsp policies session-policies policy name rule name
   session-config playback-call-settings
config vsp dial-plan dial-prefix entryName session-config
   playback-call-settings
config vsp dial-plan route name session-config playback-call-settings
config vsp dial-plan source-route name session-config
   playback-call-settings
config vsp session-config-pool entry name playback-call-settings
```

Properties

Property name	Description
playback-last-call {enabled | disabled}	Configures the system to playback the last call between the To/From call pair instead of initiating a new call. Note that the anchor and record properties must be enabled in the media object of the default session configuration for recording to take place. **Example: set playback-last-call enabled** The default setting is **disabled**.

csta-settings

Purpose

Provides CSTA-to-OCI, -OCS, or - communications for enterprises using a BroadWorks, Cisco, or Avaya call manager and Microsoft OCS. AA-SBC supports third-party call control (3PCC) for any phones connected to the 3PCC server. See Chapter 71, “Third-party call control server objects” for a complete description of 3PCC and information on configuring 3PCC servers.

Note that to make this application work in addition to the AA-SBC configuration, you must also point the CSTA SIP traffic at AA-SBC so that it acts as a CSTA gateway. See the *Net-Net OS-E – Session Services Configuration Guide* for more information.

Identifying the active device

Every device is mapped to a unique terminal ID. When a device logs into MOC, AA-SBC records the terminal ID. It is not uncommon for phone address to be mapped to multiple devices, and therefore, associated with multiple IDs. Because AA-SBC always selects the active device when initiating a session, there must be some logic configured to identify the active device so that only that device is used when an outbound call is made through MOC. This is useful, for example, when a home and work phone are mapped to the same URI. The **terminal-select-dial** property (in this object) configures AA-SBC to call a specified number (and play a recorded file). When a user picks up in response to the call, the device used to answer is noted as the active terminal and calls to that URI are forwarded to that active device. Note that after an ID has been established through this configuration, if a different device answers a call, AA-SBC uses the new device for the current call and then resumes use of the established device for future calls. (To change the active terminal setting, use the jtapi-control action.)

Using partitions and calling search spaces

AA-SBC allows you to use the multiple partition feature of Cisco CallManager. A single phone with a single phone number (DN) can be mapped to two or more partitions, where each partition can map to a individual line on the phone. The primary partition will indicate the line to use for outbound calls, typically this will be line 1. See the Cisco online documentation, *Partitions and Calling Search Spaces*, for complete information on Cisco partitions.

Syntax

```
config vsp default-session-config csta-settings
config vsp policies session-policies policy name rule name
   session-config csta-settings
config vsp dial-plan dial-prefix entryName session-config
   csta-settings
config vsp dial-plan route name session-config csta-settings
config vsp dial-plan source-route name session-config csta-settings
config vsp session-config-pool entry name csta-settings
```

Properties

Property name	Description
mode {none | internal | broadworks | cisco | avaya | loopback} *serverReference*	Specifies the 3PCC server type that AA-SBC is connecting to the Microsoft OCS application. By selecting a server type, AA-SBC acts as a translation device, converting CSTA traffic from that server type to a format the 3PCC server can recognize. Enter the type and a reference to the configured 3PCC server. Select either: • **none**—AA-SBC does not provide 3PCC services. • **internal**—the system acts as the PBX, resulting in phones registering with the AA-SBC device. This mode only works with phones registered directly to the AA-SBC device. • **broadworks**—AA-SBC converts CSTA traffic to either OCI or OCS traffic, depending on the **type** property setting of the referenced BroadWorks server. • **cisco**—AA-SBC converts CSTA traffic to for processing by the referenced Cisco server. • **avaya**—AA-SBC converts CSTA traffic to for processing by the referenced Avaya server. • **loopback**—AA-SBC creates a loopback session to the OCS for testing. **Example: set mode broadworks “vsp enterprise 3pcc-servers broadworks-csta-server BWocs”** The default setting is **none**.
terminal-select-dial {disabled | once-at-login *toURIexp toURIreplace fromURI file* | action-driven *toURIexp toURIreplace fromURI file*}	See Identifying the active device for information on the use of this property. In the example below, the system looks changes calls in the form of tel:+1508xxxyyyy to 1508xxxyyyy@callme.com. The result is the number the system dials to play the file. The entry in the From URI field is displayed as the caller ID. **Example: set terminal-select-dial once-at-login ^tel:(\+)?((1?508)[0-9]{7}).*$ \2@callme.com gday.wav** The default setting is **any**.

Property name	Description
lcs-transport {any | UDP | TCP | TLS *certificateReference*}	Sets the transport protocol used to communicate with the third-party server. For a secure connection and to support CSTA failover operations, set transport to TLS and include a reference to a certificate on the system. **Example: set transport tls "vsp tls certificate nnos-e.abc.com"** The default setting is **any**.
default-partition {automatic | specified *value* | derived *regEx replacement*}	Sets which partition this session configuration applies to (controls). This feature only applies to Cisco CallManager partitions; see *Partitions and Calling Search Spaces* in the Cisco online documentation for more information on partitions. Set AA-SBC to one of the following: • **automatic**—use the first address from the list retrieved from the switch. • **specified**—use the partition that you specify. • **derived**—use the partition name found in the Active Directory number, and derived by a regular expression rule. The AD number is in the form **tel:+number:partition**. Note that you can also set the default partition using the **set-default-partition** option of the jtapi-control action. The action setting overrides the values set with this property. **Example: set default-partition derived "(?ms).*<RequestSystemStatus.*tel:\+[0-9]*;(.*)PT" "\1PT"** The default setting is **automatic**. For more information regarding configuring regular expressions and replacement strings, see the section in Chapter 1, "Using regular expressions" on page 36.

sip-session-timers-settings

Purpose

Sets the values of SIP session timers as they are described in *RFC 4028, Session Timers in the Session Initiation Protocol (SIP)*. These timers establish periodic refreshes of SIP sessions, through re-INVITE and UPDATE requests, to help user agents and proxies determine whether the session is still active. In other words, it provides a keepalive functionality. Use of these timers helps prevent proxies that retain state information from needlessly keeping a call active. (This may happen if the proxy, for whatever reason, does not receive a BYE from the UA.)

There are two specific timer values that are used to derive the refresh interval:

- the session expiration, which defines the maximum length of a session
- the minimum allowed value for the session expiration.

AA-SBC always sends out the largest timer value that it knows of. For example, if a UAC sends a session expiration time that is lower than the AA-SBC configured timer, AA-SBC forwards the call with its own setting. If the session expiration set by the UAC is higher, AA-SBC uses the UAC setting. If the session expires, the configurable **action** determines the AA-SBC response.

The timer values that you set with this object apply to all calls that are processed through the AA-SBC device.

Syntax

```
config vsp default-session-config sip-session-timers-settings
config vsp policies session-policies policy name rule name
   session-config sip-session-timers-settings
config vsp dial-plan dial-prefix entryName session-config
   sip-session-timers-settings
config vsp dial-plan route name session-config
   sip-session-timers-settings
config vsp dial-plan source-route name session-config
   sip-session-timers-settings
config vsp session-config-pool entry name sip-session-timers-settings
```

Properties

Property name	Description
admin {enabled | disabled}	Specifies whether the SIP session timers configuration is in use or not. If **enabled**, the system adds a Session-Expires and a Min-SE header to the INVITE request. **Example: set admin enabled** The default setting is **disabled**.
preferred-refresher {UAC | UAS}	Specifies which side of the connection performs the refreshes (send the re-INVITE and UPDATE requests). Select either UAC or UAS. If the selected device is not configured to act as a refresher, this property resets that configuration to enable refreshing (assuming the device implements the RFC). **Example: set preferred-refresher UAS** The default setting is **UAC**.
session-expires *seconds*	Specifies the duration of the session. This is the maximum time allowed between session refresh requests before a session times out. The RFC notes that you can, but should not, set a value of less than 1800 (30 minutes), as it causes excessive messaging. **Example: set session-expires 2700** Enter a value in the range of 90 to 1,000,000; the default setting is **1800** seconds.

Property name	Description
min-se *seconds*	Specifies the minimum allowed value for the session expiration interval. This lower floor is the fastest refresh rate a proxy servicing a request can require. A proxy can raise but not lower this minimum. **Example: set min-se 180** Enter a value in the range of 90 to 1,000,000; the default setting is **90** seconds.
action {terminate | disconnect | nothing}	Specifies the action the system should take when the SIP session timers have expired. • **terminate**—the system immediately removes all internal resources dedicated to the call. • **disconnect**—the system sends a BYE in an attempt to gracefully close the call before terminating it. • **nothing**—the system ignores the timer expiration and keeps the session alive until it receives a BYE. **Example: set action disconnect** The default setting is **terminate**.

routing-settings

Purpose

Provides mechanisms for filtering the routing table. For example, when configured and a policy match occurs, AA-SBC can send traffic to interfaces with the same geolocation setting. Or, AA-SBC can control the egress interface using the routing and classification tag system. (See Tag-based route selection for a complete description of how AA-SBC classifies traffic.) Otherwise, AA-SBC uses the entire routing table when identifying interface choices.

➡ **Note:** The preferred method for creating virtual firewalls is by using routing tags and VLANs. For sample configurations that illustrate VLANs, overlapping IP addresses, and virtual firewalls, see the *Net-Net OS-E – System Administration Guide*.

Syntax

```
config vsp default-session-config routing-settings
config vsp policies session-policies policy name rule name
   session-config routing-settings
config vsp dial-plan dial-prefix entryName session-config
   routing-settings
config vsp dial-plan route name session-config routing-settings
config vsp dial-plan source-route name session-config routing-settings
config vsp session-config-pool entry name routing-settings
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables this route setting configuration. **Example: set admin disabled** The default setting is **enabled**.
geolocation *value*	Assigns a numeric to match on for the ip interface geolocation property. When a match occurs, the system forwards traffic to an interface with a matching geolocation. If you set the value to 0, the system uses the whole routing table when selecting an interface. **Example: set geolocation 10** The default setting is **0**.

Property name	Description
ingress-classification-tag *string*	Classifies (associates) incoming traffic with the configured tag. If you configure this property, you must configure the **egress-classification-tag** property (below) as well. The configured ingress-tag must match a configured ip routing-tag. The routing-tag identifies the groups of interfaces and routes that are available for the ingress-tag. See Tag-based route selection for a complete description. You can also configure a **classification-tag** through the ip interface object. If this property is configured in both places, the session-config setting takes precedence. Note that this tag is case-sensitive. **Example: set ingress-classification-tag E911** There is no default setting.
egress-classification-tag *string*	Sets an egress classification tag that is used to select the outgoing interface. That tag must then be associated with an ip **routing-tag**, which controls the available egress interfaces and routes. See Tag-based route selection for a complete description. You can also configure a **classification-tag** through the ip interface object. If this property is configured in both places, the session-config setting takes precedence. Note that this tag is case-sensitive. **Example: set ingress-classification-tag E911** There is no default setting.

pre-call-authorization

Purpose

Configures the call access mechanism in AA-SBC. When configured and enabled, AA-SBC terminates matching calls at the box. The system then collects the DTMF digits it receives until either the timeout expires, the verify signal is received, or maximum number of digits has been reached. After digit collection is terminated, AA-SBC compares the digits to its list of authorized numbers for that caller. If the collected digit string matches an entry in that list, AA-SBC plays a success message, if configured, and completes the call. If the collected digit string does not match an entry in that list, AA-SBC plays the failure message, if configured, and terminates the call. In addition, while digits are being collected, if AA-SBC encounters the configured “cancel” digit, the system terminates the call. If the “restart” digit is enabled and encountered, all previously entered digits are removed, and collection continues.

Syntax

```
config vsp default-session-config pre-call-authorization
config vsp policies session-policies policy name rule name
session-config pre-call-authorization
config vsp dial-plan dial-prefix entryName session-config
pre-call-authorization
config vsp dial-plan route name session-config pre-call-authorization
config vsp dial-plan source-route name session-config
pre-call-authorization
config vsp session-config-pool entry name pre-call-authorization
```

Properties

Property name	Description
admin {enabled | disabled}	Sets the administrative status of the configuration entry for terminating incoming calls at the box. **Example: set admin disabled** The default setting is **disabled**.
greeting-message *path*	Sets the message that is played to incoming calls if they have been terminated. Enter a path name to the WAV file containing the message. **Example: set greeting-message /cxc_common/preauth.wav** There is no default setting.
max-number-of-digits *integer*	Specifies the maximum number of digits AA-SBC collects before it begins evaluating the string against the authorized list (set with the **authList** property). **Example: set max-number-of-digits 10** Enter a value between 1 and 16; the default setting is **7** digits.
verify {disabled | enabled *telephoneKey}*	Configures a signal (telephone key pad key) that AA-SBC waits for before it begins evaluating the string against the authorized list (set with the **authList** property). Use this to allow, for example, variable length PINs. Configure a pound (#) symbol to indicate end of collection and end PIN entries with a pound. **Example: set verify enabled Pound** The default setting is **disabled**.
cancel {disabled | enabled *telephoneKey}*	Configures forced termination of a call. When **enabled**, sets a signal (telephone key pad key) that, if encountered in DTMF collection, forces AA-SBC to terminate the call. **Example: set cancel enabled Star** The default setting is **disabled**.

Property name	Description
restart {disabled | enabled *telephoneKey}*	Configures a collection restart. When **enabled**, sets a signal (telephone key pad key) that, if encountered in DTMF collection, forces AA-SBC to wipe whatever digits have been collected and restart collection. **Example: set restart enabled Flash** The default setting is **disabled**.
inter-digit-timeout *milliseconds*	Configures a length of time AA-SBC waits after button pushing has stopped and before it begins evaluating the string against the authorized list (set with the **authList** property). The inter-digit timer is reset every time a digit is received. When it times out, the preceding digits will be verified. This is used, for example, when entering a PIN. The number is not verified until the caller stops entering; after the inter-digit time out expires AA-SBC checks the string. **Example: set inter-digit-timeout 5000** Enter a value from 500 to 60,000; the default setting is **2500** milliseconds.
success-message *path*	Sets the message that is played to the caller indicating that the call attempt has been successful. This is played if the collected digits match an entry in the authorized list (set with the **authList** property). Enter a path name to the WAV file containing the message. **Example: set success-message /cxc_common/callSuccess.wav** There is no default setting.

Property name	Description
failure-message *path*	Sets the message that is played to the caller indicating that the call attempt was not successful. This is played if the collected digits do not match an entry in the authorized list (set with the **authList** property). Enter a path name to the WAV file containing the message. **Example: set failure-message /cxc_common/callFail.wav** There is no default setting.
authList *digitString*	Adds strings of digits to a list of authorized calling numbers. AA-SBC uses this list to match against the collected digits and determine whether a call is allowed. Re-execute the command to enter multiple strings. **Example: set authList 011** There is no default setting.

third-party-call-control

Purpose

Configures call control, allowing AA-SBC or a CSTA client to control (become the third party) in a call. Specifically, this object controls the WAV files that AA-SBC should play and the external status events reported to an external server for calls created by the AA-SBC device. The third-party call controller (3PCC) functionality is used, for example, to enable the interworking of a uaCSTA with the Broadworks Open Client Interface. AA-SBC converts between the two call control protocols. Phone control can be integrated, for example, into Microsoft Office applications using the Phone Controls interface. This object can also be enabled in certain situations involving LCS/Sametime interworking and other advanced AA-SBC applications. In all cases, it should only be enabled at the direction of Technical Support.

When AA-SBC functions as a 3PCC device by initiating communications to each endpoint in the session, this object configures the specific WAV file(s) to play in response to the state of the call destination. Specifically, when AA-SBC receives an instruction from the CSTA client to establish a call, it first makes a call to the originator of the call and then to the destination. The destination responds with call progress information. If that information indicates that the phone is ringing, and the **ringback-file** property is configured, AA-SBC plays the specified file. If the phone is busy or set to appear so, AA-SBC plays any configured **busy-file** recording.

Note: You must set the **admin** property of this object to **enabled** if you are implementing a Sametime-to-LCS federation. If you are running Sametime-to-Sametime or LCS-to-LCS, the **admin** property must be **disabled**.

Assuring pre-call announcements after failover

When the **pre-call-announcement** property is set, AA-SBC plays a WAV file for the caller prior to the call connecting. If you set this property and want to ensure the pre-call announcement is played in the event that the master box fails over to a backup box, you must copy files to the backup boxes using the file mirror service. To do so:

1. Enable the file-mirror master service and configure a **file-mirror-directory**. List all boxes as possible hosts with the **host-box** property. For example:

```
config file-mirror> set admin enabled
config file-mirror> set file-mirror-directory /cxc_common/mirror
config file-mirror> set host-box cluster\box 1
config file-mirror> set host-box cluster\box 2
config file-mirror> set host-box cluster\box 3
```

2. Upload the pre-call announcement WAV file to the configured mirror directory.

3. Execute the file-mirror-service **make-available** action. For example:

```
NNOS-E> file-mirror-service make-available /cxc_common/mirror/
   announcement.wav
```

By executing the action after uploading files to the file mirror directory, these files will be available to master and drone in the event of failover.

Syntax

```
config vsp default-session-config third-party-call-control
config vsp policies session-policies policy name rule name
   session-config third-party-call-control
config vsp dial-plan dial-prefix entryName session-config
   third-party-call-control
```

```
config vsp dial-plan route name session-config
   third-party-call-control
config vsp dial-plan source-route name session-config
   third-party-call-control
config vsp session-config-pool entry name third-party-call-control
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables the ability to control externally originated calls. (If the system is the call originator, this setting is ignored.) **Example: set admin enabled** The default setting is **disabled**.
status-events {originate | answer | neither | both}	Specifies when to generate status events if the system is functioning as a third-party call controller (3PCC), initiating communications to each endpoint in the session. Specifically, when the system receives an instruction from the CSTA client to establish a call, it first makes a call to the originator of the call and then to the destination. Select to send events to the external event server relating to: • **originate**—contact established with the call originator. • **answer**—contact established with the call destination. • **neither**—no status events are sent. • **both**—contact established with the call originator and destination. **Example: set status-events originate** The default setting is **both**.
handle-refer-locally {enabled | disabled}	Specifies whether the system forwards REFER messages or whether it consumes the message and generates an INVITE for the call. When an end user wants to transfer a call, the system receives a REFER message. When this property is set to **enabled** (the default), the system terminates the message, generates an INVITE, and makes a call on behalf of that call transfer. When set to **disabled**, the system forwards the call to the upstream server for handling. **Example: handle-refer-locally disabled** The default setting is **enabled**.

Property name	Description
refer-maintain-identity {true | false}	Specifies how the system handles the TO and FROM URIs when processing a REFER request. If set to **true**, the system keeps the original TO and FROM components of the URI instead of replacing them with the REFER destination. When set to **false**, the default, they are replaced. For example, suppose a third-party call is initiated from A to B, and B subsequently sends a REFER to transfer the call. If this property is set to **true**, the call appears to A as if it were still connected to B. If set to **false**, the call appears to A as if it were connected to C. **Example: set refer-maintain-identity true** The default setting is **false**.
ringback-file *filePath*	Specifies the WAV file that the system plays while the remote endpoint is ringing. If the system is the call originator, and this property is configured, the system plays the file regardless of the **admin** setting. If the call is externally originated, the ringback is generated locally by the originating phone. **Example: set ringback-file /cxc/cxc_common/holdRinging.wav** There is no default file setting.
busy-file *filePath*	Specifies the WAV file that the system plays when the remote endpoint is set to do-not-disturb or is busy. If the system is the call originator, and this property is configured, the system plays the file regardless of the **admin** setting. If the call is externally originated, the busy signal is generated locally by the originating phone. **Example: set busy-file /cxc/cxc_common/busy1.wav** There is no default file setting.

Property name	Description
pre-call-announcement *filePath*	Specifies the WAV file that the system plays for the caller when this property is set. The callee does not hear the file (the file is played before the connection to the destination is attempted). In contrast, use the introduction property of the media object to play an recording for both sides. **Example: set pre-call-announcement /cxc/cxc_common/youhavewon.wav** There is no default file setting.
terminate-after-pre-call-announcement {enabled | disabled}	Specifies whether to end a call once the pre-call announcement has been played. When **enabled**, the system plays the file specified with the **pre-call-announcement** property and then terminates the call. **Example: set terminate-after-pre-call-announcement enabled** The default setting is **disabled**.
handle-replaces-locally {enabled | disabled}	Specifies whether the system forwards INVITEs with a Replaces header or whether it consumes the message and generates a new INVITE for the call. When an endpoint transfers a call, AA-SBC receives a REFER message. When this property is **enabled**, the system terminates the message and completes the replacement locally. When **disabled**, the system forwards the call to the upstream server for handling. **Example: set handle-replaces-locally enabled** The default setting is **disabled**.
delayed-ack {enabled | disabled}	Specifies when the system sends an ACK in response to an originating call leg. When **enabled**, the system does not send an ACK to the originating call leg until it receives a 200/OK from the answering call leg. When **disabled**, the system sends an ACK immediately, prior to receiving a response from the destination. **Example: set delayed-ack enabled** The default setting is **disabled**.

Property name	Description
include-reason-in-bye {enabled | disabled}	Specifies whether to include the optional Reason header in a BYE request. Enable this to resolve the Heterogeneous Error Response Forking Problem (HERFP), as stated in RFC 3326. The Reason field helps in properly updating requests when proxy servers forward requests to multiple contacts associated with a call. When **enabled**, the default, the Reason header is included. **Example: set include-reason-in-bye disabled** The default setting is **enabled**.
always-apply-req-uri-spec {enabled | disabled}	Specifies when to include the settings of the request-uri-specification. When **enabled**, the system applies the settings to the initial INVITE as well as any re-INVITEs. When **disabled**, the system only applies the settings on the initial INVITE. **Example: set always-apply-req-uri-spec disabled** The default setting is **enabled**.
media-shuffle {enabled | disabled}	Specifies whether to generate SDP locally in H.323-to-SIP calls when the H.323 endpoint is not in fast-start mode. (Fast-start mode is the option of having the H.323 endpoint provide media information in the initial call setup.) In a non-fast-start case, the system has no media information when it sends out an INVITE. When this property is **enabled**, the system generates some SDP initially. When it receives the real media information from the H.323 side, the system then reinvites the SIP endpoint with the correct information. When media-shuffle is **disabled**, the system sends the INVITE with no SDP. **Example: set media-shuffle disabled** The default setting is **enabled**.
park-incoming-calls {enabled | disabled}	Allows establishment of an inbound 3PCC session. When **enabled**, any incoming call is answered by the system (similar to the results of the call-control **park** action). Use this, for example, in conjunction with the **pre-call-announcement** property. **Example: set park-incoming-calls enabled** The default setting is **disabled**.

Property name	Description
parked-call-greeting *filePath*	Specifies the WAV file that the system plays for the caller when the **park-incoming-calls** property is set to enabled. The callee does not hear the file (the file is played before the connection to the destination is attempted). In contrast, use the introduction property of the media object to play a recording for both sides. **Example: set parked-call-greeting /cxc/cxc_common/park.wav** There is no default file setting.
terminate-after-greeting {enabled | disabled}	Specifies whether AA-SBC terminates a call after playing the file specified with the **parked-call-greeting** property. The **park-incoming-calls** property must be set to enabled for this property to apply. **Example: set terminate-after-greeting enabled** The default setting is **disabled**.
terminate-update-locally {enabled | disabled}	Specifies whether the system responds locally to UPDATE messages. When this is **enabled**, if an endpoint sends an UPDATE message, the system responds to the update with a 200 OK and the SDP from the remote endpoint. If **disabled**, the system forwards the UPDATE to the remote endpoint. **Example: set terminate-update-locally enabled** The default setting is **disabled**.
terminate-reinvite-locally {enabled | disabled}	Specifies whether the system responds locally to REINVITE messages. When this is **enabled**, if an endpoint sends a REINVITE message, the system responds to the reinvite with a 200 OK and the SDP from the remote endpoint. If **disabled**, the system forwards the REINVITE to the remote endpoint. **Example: set terminate-reinvite-locally enabled** The default setting is **disabled**.

Property name	Description
forking-early-media-inhibit {enabled | disabled}	Specifies whether to strip SDP from a provisional response. Use this in some cases when using sequential or parallel forking to prevent an endpoint from being swamped by provisional response content. The saved SDP is later inserted into the 200 OK (unless more current SDP is available in the 200 OK), allowing the calling phone to receive just one SDP. **Example: set forking-early-media-inhibit enabled** The default setting is **disabled**.
forward-unresolved-replaces {enabled | disabled}	*Secondary property.* Specifies whether to forward a 481(Call Leg Not Found) message when the system cannot find the call leg information in the Replaces header. In an attended transfer, A calls B, and then B calls C, makes the connection, and completes the transfer. To do so, B sends a REFER to C. Included in that REFER is a Replaces header, with information identifying the call-leg from A to B (which will be replaced). As a B2B, the system must convert the identifying information in the call leg on the B side to that which represents the call-leg on the A side. When this property is **disabled**, if the system cannot find the call-leg in the Replaces header it responds with a 481 to the system receiving the REFER. If **enabled**, the system forwards the REFER to C without the conversion described. **Example: set forward-unresolved-replaces enabled** The default setting is **disabled**.
extract-refer-to-header-spec {enabled | disabled}	*Secondary property.* Specifies whether to use the fields of the REFERTO header. When **enabled**, AA-SBC extracts the header specifications from the REFERTO header and includes them in the resulting INVITE. **Example: set extract-refer-to-header-spec enabled** The default setting is **disabled**.

Property name	Description
reinvite-preserve-media {enabled | disabled}	*Secondary property.* Specifies whether to re-use SDP. When set to **enabled**, in cases where an REINVITE is received and there is no SDP, AA-SBC will use the previous SDP. **Example: set reinvite-preserve-media enabled** The default setting is **disabled**.
media-forward {enabled | disabled}	*Secondary property.* Specifies whether AA-SBC can respond to forwarded requests from the NICE media server. When this property is **enabled**, the system treats all INVITEs as requests from the NICE media server. (If the INVITE is not from the NICE media server, the system will reject the request after scanning and not finding NICE-specific content.) **Example: set media-forward enabled** The default setting is **disabled**.
track-to-user {enabled | disabled}	*Secondary property.* Allows the NICE media server to request forwarding sessions based on the user listed in the TO URI. **Example: set track-to-user enabled** The default setting is **disabled**.
force-retrieve-on-delayed-offer-while-held {enabled | disabled}	*Secondary property.* Specifies system behavior when an INVITE with no SDP is received for a call on hold. When enabled, if an INVITE with no SDP is received, it is treated as a receive request and the call is taken off hold. When disabled, call status does not change from that event. **Example: set force-retrieve-on-delayed-offer-while-held enabled** The default setting is **disabled**.

Property name	Description
reinvite-delayed-offer-wait-on-ack {enabled | disabled}	*Secondary property.* Specifies how AA-SBC responds to a REINVITE with no SDP. When **enabled**, AA-SBC postpones re-inviting the remote end (accepting the REINVITE locally) until receiving the SDP contained in the ACK. The default setting is **disabled.** **Example: set reinvite-delayed-offer-wait-on-ack enabled** The default setting is **disabled**.
use-183-for-ringing-with-sdp {enabled | disabled}	*Secondary property.* Specifies the SIP code that AA-SBC forwards with messages it receives that contain early media. If this property is **disabled**, the system forwards the message with a 180 (Ringing) code. If **enabled**, the system changes the code to 183 (Session Progress) when forwarding the message. **Example: set use-183-for-ringing-with-sdp enabled** The default setting is **disabled**.
strip-require-100-rel {enabled | disabled}	*Secondary property.* Specifies whether the “Require: 100rel” line is stripped from responses. When enabled the line is removed from forwarded responses. **Example: set strip-require-100-rel enabled** The default setting is **disabled**.

Property name	Description
forward-all-parallel-provisional-responses {enabled | disabled}	*Secondary property.* Sets how provisional responses are handled when parallel forking is enabled. Parallel forking results in a single incoming INVITE generating multiple simultaneous outgoing INVITEs, which can cause issues regarding which audio stream to apply if multiple provisional responses are returned. When this property is **enabled**, if a provisional response is received during a parallel forking attempt, every 18*x* response received prior to one of the outgoing calls completing successfully is forwarded to the originating agent. If this property is **disabled**, provisional responses are only forwarded when received from the first server on the list of destinations being attempted. **Example: set forward-all-parallel-provisional-responses enabled** The default setting is **disabled**.
include-id-in-refer-notify {enabled | disabled}	*Secondary property.* Specifies whether, when a NOTIFY is sent reporting the final status of a REFER call, AA-SBC includes an ID field in the NOTIFY. When **enabled**, AA-SBC includes the ID. **Example: set include-id-in-refer-notify enabled** The default setting is **disabled**.
notify-dtmf-event-if-allowed {enabled | disabled}	*Secondary property.* Sets whether the system sends a NOTIFY event in response to receiving DTMF. To use this property, the remote user agent must allow KPML or telephone events. When AA-SBC receives an Allow-Events header, and this property is **enabled**, upon receiving DTMF the system sends a NOTIFY event with the RFC 2833 representation of the received DTMF. **Example: set notify-dtmf-event-if-allowed enabled** The default setting is **disabled**.

Property name	Description
terminate-hold-retrieve-locally {enabled | disabled}	When this property is enabled, if a re-INVITE is received with an SDP that indicates it is either a hold or retrieve request, the AA-SBC accepts the re-INVITE locally with the SDP acknowledging the hold or retrieve and the message is not forwarded. When this property is disabled, re-INVITE messages with an SDP that indicates hold or retrieve receive no special treatment. **Example: set terminate-hold-retrieve-locally enabled** The default setting is **disabled**.
reinvite-originator {enabled | disabled}	When enabled, the AA-SBC reinvites the original UAC after the call is initially set up. **Example: set reinvite-originator enabled** The default setting is **disabled.**
skip-shuffle-complete-if-anchored {enabled | disabled}	When enabled, no reinvite is sent forwarding the SDP contained in the ACK for calls with anchored media. **Example**: **set skip-shuffle-complete-if-anchored enabled** The default setting is **disabled**.
forward-302-division-header {enabled | disabled}	*Secondary property*. When enabled, if a 302 Redirected response with a Diversion: header is received by the AA-SBC, the Diversion: header is forwarded in the response. **Example**: **set forward-302-diversion-header disabled** The default setting is **enabled**.

Property name	Description
inhibit-shuffle-update	Controls whether a re-INVITE message is generated and sent to the destination server to update the SDP (as some user agents initiate SIP calls using an INVITE without an SDP.) Normally, when the **media-shuffle** property is enabled, the SDP is generated locally for the outgoing call, and the answer SDP from the destination will be returned to the originator as the offer SDP. After the originator responds with the answer SDP, a reinvite will be generated to the destination to update the SDP. When **inhibit-shuffle-update** is set to enabled, this re-INVITE will not be sent to update the SDP. The inhibit-shuffle-update property setting is only valid when media-anchoring is enabled. If inhibit-shuffle-update is enabled without media anchoring enabled, audio problems will result. **Example**: **set inhibit-shuffle-update enabled** The default setting is **disabled** .
refer-notify-100-trying	Controls the behavior of the AA-SBC when a REFER message is received and the referrer disconnects before the resulting transfer call has completed. **`Example: set refer-notify-100-trying test`**
forward-302-diversion-header {enabled | disabled}	When enabled, if a 302 Redirected response with a Diversion: header is received by the AA-SBC, the Diversion: header is forwarded in the response. **Example**: **set forward-302-diversion-header disabled** The default setting is **enabled**.

Property name	Description
media-forward-reference-direction	Identifies which leg of a call is to the call-center PBX, mapping the Rx and Tx streamsx to match the NICE equipment Rx and Tx streams. The following are valid options: • in-leg • out-leg **Example**: **set media-forward-reference-direction in-leg** The default setting is **out-leg**.
inhibit-100-trying-for-reinvite	When enabled, the AA-SBC does not send out a 100 Trying when it receives a re-INVITE. When disabled, the AA-SBC does send out a 100 Trying in response to a re-INVITE. Example: **set inhibit-100-trying-for-reinvite disabled** The default setting is **enabled**.

uui-header

Purpose

Makes modifications to the user-to-user information (UUI) header. The UUI can be used for passing the universal call ID (UCID) and other session information to the NICE media server. If this object is configured, when AA-SBC receives a SIP message it checks for the UUI header and applies the action defined with the **replace-existing-header** property. If no UUI header exists, AA-SBC creates one.

Syntax

```
config vsp default-session-config uui-header
config vsp policies session-policies policy name rule name
   session-config uui-header
config vsp dial-plan dial-prefix entryName session-config uui-header
config vsp dial-plan route name session-config uui-header
config vsp dial-plan source-route name session-config uui-header
config vsp session-config-pool entry name uui-header
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables this header manipulation configuration entry. **Example: set admin enabled** The default setting is **disabled**.
node-id *integer*	Specifies the number to be written to the UUI header, identifying the system to the media server. This is an internal integer specific to AA-SBC; enter any value between 1 and 65535. **Example: set node-id 123** The default setting is **0**.
replace-existing-header {enabled | disabled}	Specifies the action the system should take if there is an existing UUI header. If **disabled**, the system appends the modified UUI header to the existing one. If **enabled**, the system replaces the existing header with the new, modified version. **Example: set replace-existing-header enabled** The default setting is **disabled**.

handle-response

Purpose

Assigns an action to a matching response code. If AA-SBC receives a response to a SIP request (typically a UAS) that is within the 400-999 range, you can specify how AA-SBC responds (forwarding the response message or re-sending the INVITE to a different peer or route). The purpose of this feature is to determine the destination to which AA-SBC forks a call.

AA-SBC operates in the following manner. If:

- the **handle-response** property is configured for the session-config and there is a matching response code, AA-SBC takes the configured action. If the response code does not match, AA-SBC uses the default action (try-next-peer).
- the **handle-response** property is not configured for the session-config, AA-SBC uses the handle-response setting in the server configuration (if it exists).

When AA-SBC receives a the specified response code from a call, it takes one of the following actions:

- **try-next-peer**—AA-SBC forwards the message to the next server within a route.
- **try-next-route**—AA-SBC forwards the message to the route that is the next most-specific.
- **forward**—AA-SBC returns the response to the originator of the message.

A call can match more than one route, and each route may have more than one destination. For example, a call may route to destinations A and B. Destination A may have routes A1, A2, and A3. Destination B may have routes B1 and B2. If the current destination is A1, and a handle-response code match occurs with a setting of **try-next-peer**, AA-SBC forwards the message to A2. If the setting is **try-next-route**, AA-SBC forwards the message to B1.

Syntax

```
config vsp default-session-config handle-response
config vsp policies session-policies policy name rule name
   session-config handle-response
config vsp dial-plan dial-prefix entryName session-config
   handle-response
config vsp dial-plan route name session-config handle-response
config vsp dial-plan source-route name session-config handle-response
config vsp session-config-pool entry name handle-response
```

Properties

Property name	Description
`entry` *`code`* `{try-next-peer | try-next-route | forward}`	Specifies the action the system should take when it receives a specific response code from a call matching this session-config. Enter a code, and set a handling pattern: • **try-next-peer**—the system forwards the message to the next server for a route. • **try-next-route**—the system forwards the message to the route that is the next most-specific. • **forward**—the system returns the response to the originator of the message. **Example:** set handle-response 404 try-next-route There is no default value. Enter a response code between 400 and 999. The default setting for the handling pattern is **try-next-peer**.

handle-publish

Purpose

Sets whether events are sent to a third-party server via .

Syntax

```
config vsp default-session-config handle-publish
config vsp policies session-policies policy name rule name
   session-config handle-publish
config vsp dial-plan dial-prefix entryName session-config
   handle-publish
config vsp dial-plan route name session-config handle-publish
config vsp dial-plan source-route name session-config handle-publish
config vsp session-config-pool entry name handle-publish
```

Properties

Property name	Description
`third-party-interface {none | }`	Sets whether events are sent to a third-party server via . **Example:** set third-party-interface The default value is **none**.

3GPP

Purpose

Configures 3rd Generation Partnership Project (3GPP) systems.

Syntax

```
config vsp default-session-config 3GPP
config vsp policies session-policies policy name rule name
   session-config 3GPP
config vsp dial-plan dial-prefix entryName session-config 3GPP
config vsp dial-plan route name session-config 3GPP
config vsp dial-plan source-route name session-config 3GPP
config vsp session-config-pool entry name 3GPP
```

Properties

None

Rx

Purpose

Configures communication between AA-SBC and the Camiant Policy Server (3GPP Rx). The Camiant Policy Server applies business rules that determine which and when customers, tiers, and/or applications receive bandwidth priority. AA-SBC notifies the Camiant system about media flows so that the Camiant system can then apply policy for resource allocation (or example, reserving bandwidth or setting up QoS and forwarding rules). Communication between the two system uses a 3GPP Diameter-based Rx interface.

This object sets destination Diameter servers and configures what AA-SBC does in response to an RAR received from the Camiant (3GPP Rx) box. Using this configuration AA-SBC is not authenticating a message, but is instead authorizing a service and reserving bandwidth for it. The Camiant box then sends the Re-Authorization Requests (RARs) when there is a network event that requires it and AA-SBC acts according to the mapped response.

Syntax

```
config vsp default-session-config 3GPP Rx
config vsp policies session-policies policy name rule name
   session-config 3GPP Rx
config vsp dial-plan dial-prefix entryName session-config 3GPP Rx
config vsp dial-plan route name session-config 3GPP Rx
config vsp dial-plan source-route name session-config 3GPP Rx
config vsp session-config-pool entry name 3GPP Rx
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables the Rx configuration. When **enabled**, the system forwards Rx requests to the configured server and takes the configured actions on RARs. When **disabled**, the system does not send messages to or receive them from the Camiant server. **Example: set admin enabled** The default setting is **disabled**.

Property name	**Description**
diameter-group *diameterReference*	Specifies a previously configured diameter-group to which the system sends Rx requests. **Example: set diameter-group “vsp diameter-group 3gppGroup”** There is no default setting.
re-auth-action *cause action*	Specifies the action the system takes on Re-Authorization Request (RAR) messages it receives from the Camiant (3GPP Rx) box. The Camiant server sends a cause, and the system takes the action mapped to that cause with this property. Select a recognized reauthorization cause: • service-info-request • charging-correlation-change • loss-of-bearer • recovery-of-bearer • release-of-bearer • establishment-of-bearer Assign one of the following actions: • **none**—take no action. • **re-authenticate**—attempt to reauthenticate the message by sending a new Diameter Authentication-Authorization-Request. • **disconnect**—disconnect the call. **Example: set re-auth-action service-info-request re-authenticate** The default setting configures the system to **disconnect** the call for a **release-of-bearer** RAR message.

response-translation-settings

Purpose

Maps a new status code and, optionally, phrase to a received code.

Syntax

```
config vsp default-session-config response-translation-settings
config vsp policies session-policies policy name rule name
   session-config response-translation-settings
config vsp dial-plan dial-prefix entryName session-config
   response-translation-settings
config vsp dial-plan route name session-config
   response-translation-settings
config vsp dial-plan source-route name session-config
   response-translation-settings
config vsp session-config-pool entry name
   response-translation-settings
```

Properties

Property name	Description
entry *statusCode newStatusCode* [*reason*] [*newReason*]	Sets a new numeric to a reason code. Specify a code number and the replacement number to use instead. Optionally, you can specify the reason phrase and a replacement phrase. If you specify both the code and phrase, the incoming message must contain both for the replacement to take place. **Example: set entry 503 200 “Service Unavailable” “Service Delayed”** The default value is **none**.

accounting-data

Purpose

Adds a custom data field to the accounting record. Use this object to define the content of the field. AA-SBC supports several predefined selections for use with the **entry** property. (You can also add text string values.) To enter a predefined value, precede the letter with slash (and quotation marks in the CLI). For example, “\b” to add a box identifier. The following are the predefined selections for use with this object.

Use	**For...**
\b	box identifier
\s	source LNP
\e	destination LNP
\d	digest-realm
\v	diVersion header
\p	p-charging-vector
\c	cluster name
\r	RADIUS caller ID
\z	connected
\y	scan-time
\x	file-time
\w	play-time
\u	disconnect-reason
\t	final-code
\o	action-id

Syntax

```
config vsp default-session-config accounting-data
config vsp policies session-policies policy name rule name
   session-config accounting-data
config vsp dial-plan dial-prefix entryName session-config
   accounting-data
config vsp dial-plan route name session-config accounting-data
config vsp dial-plan source-route name session-config accounting-data
config vsp session-config-pool entry name accounting-data
```

Properties

Property name	Description
entry *tag* [*value*]	Specifies the content of the field added to the accounting record, in the format *tag*=*value*. Use this property, for example, to add a source based on calls matching this session configuration. Use the predefined elements (see the Purpose) to extract data from the call SIP headers. **Example: set entry cluster-name "\c"** There is no default setting.
post-process-expression *regExp replacement*	Configures a regular expression, and replacement text, to run against the TO and FROM fields of the CDR. Use post processing to, for example, remove unwanted commas that may appear as the result of data that was imported from a CSV file. The following example turns a comma into a dash. **Example: set post=process-expression (.*0, (.*) "\1 - \2"** There is no default setting. For more information regarding configuring regular expressions and replacement strings, see the section in Chapter 1, "Using regular expressions" on page 36.
custom-data-grouping-string	The characters used to associate custom data tags and values. **Example: set custom-data-grouping-string *** The default setting is **=**.
custom-data-delimiter	The characters used to separate group data entries. **Example: set custom-data-delimiter *** The default setting is **;**.

codec-specific-parameters

Purpose

Adds an a=fmtp line to SDP. Certain endpoints require CODEC parameters to be specified; this object allows you to add the necessary information. Note that if SDP arrives with the a=fmtp line specified for a CODEC set with this object, the original line remains. Use this object in conjunction with the sdp-regeneration object to override the line. First, use **sdp-generation** to remove the received a=ftmp line from SDP, then use this object to add the line back in with the desired parameters.

Syntax

```
config vsp default-session-config codec-specific-parameters
config vsp policies session-policies policy name rule name
   session-config codec-specific-parameters
config vsp dial-plan dial-prefix entryName session-config
   codec-specific-parameters
config vsp dial-plan route name session-config
   codec-specific-parameters
config vsp dial-plan source-route name session-config
   codec-specific-parameters
config vsp session-config-pool entry name codec-specific-parameters
```

Properties

Property name	Description
codec-parameters *codec parameters*	Specifies the CODEC and the parameter to apply to it in the a=fmtp line of SDP. **Example: set codec-parameters g729 annexb=yes** There is no default setting.

inbound-header-settings

Purpose

The inbound-header-settings configuration object allows you to set fields to remove and/or replace header settings in the SIP headers for inbound traffic.

Syntax

```
config vsp policies session-policies policy default rule
   session-config inbound-header-settings
```

Properties

Property name	Description
pAssert-mode [enabled | disabled]	*Secondary property.* Sets whether or not to strip the number in the P-Asserted-Identity field from the SIP header. When enabled, the AA-SBC replaces the value in the From field with the value from the P-Asserted-Identity field for the outbound call leg. (Note that the AA-SBC maintains the original From field value in the Contact field.) EXAMPLE: **set passert-mode enabled** The default setting is **disabled**.
header-to-strip	*Secondary property.* Configures the AA-SBC to strip the value of the specified field. Enter a SIP header field name. EXAMPLE: **set header-to-strip sip1**
allowed-header	Sets the SIP headers that should be explicitly allowed to remain in the SIP message. You can enter any number of header names by re-executing the command. EXAMPLE: **set allowed-header header1**

Property name	Description
blocked-header	Sets the SIP headers that should be explicitly removed from the SIP message. You can enter any number of header names by re-executing the command. EXAMPLE: **set blocked-header header5**
apply-allow-block-to	Sets whether the allow and block properties of this object apply to request messages, response messages, or both. The following are valid apply-allow-block-to values: requests—apply to requests only responses—apply to responses only requests-and-responses—apply to requests and responses **Example: set apply-allow-block-to responses** The default setting is **requests-and-responses**.

header-normalization

Purpose

The header-normalization object alters the user portion of the specified header.

Syntax

```
config vsp policies session-policies policy default rule
   session-config inbound-header-settings header-normalization
```

Properties

Property name	Description
admin [enabled | disabled]	Enable or disable header-normalization. **Example: set admin disabled** The default setting is **enabled**.
destination	Specifies the header to be created or modified by the properties in this object. That is, the AA-SBC modifies this URI with the data from the source. **Example: set destination P-Asserted-Identity**
value	Specifies the field in the specified destination URI to overwrite. The following are valid values: -none—no normalization applied -prepend—prepend string -prepend-to—prepend string to certain length -strip-off—strip off N characters- -strip-off-to—strip off prefix to certain length -replace-prefix—replace prefix with a different prefix -replace-withqQreplace with a different name -append—append extension number **Example: set value strip-off** The default setting is **none**.

Property name	Description
apply-to-methods	Specifies the message type to which the system applies header value changes. The AA-SBC then changes the specified URI according to the settings of the header and destination properties of this object. When you modify this value, the AA-SBC overwrites the current settings with only the message types you specify. Enter multiple types separated by a plus sign (+) with no spaces. The following are valid values: -INVITE -REFER -MESSAGE -INFO -OPTIONS -REGISTER -SUBSCRIBE -NOTIFY -PUBLISH -ACK -BYE -CANCEL -PRACK -UPDATE -SERVICE -PING **Example: set apply-to-methods publish+refer+info** The default setting is **INVITE**.

Property name	Description
apply-to-responses	Specifies whether to apply header value changes to SIP requests or requests and responses. If you enter a value of **yes**, you must include the response-code. The following are valid values: no—do not apply to responses (requests only) yes—apply to responses of this type **Example: set apply-to-responses no**. The default setting is **no**.
session-persistent [enabled | disabled]	Specifies to which messages in a session the AA-SBC should apply changes made with this object. When enabled, the AA-SBC applies any TO, FROM, or REQUEST URI changes to the first and all subsequent messages in a session. When disabled, the system applies the changes only to the first message in the session. **Example: set session-persistent enabled** The default setting is **disabled**.

altered-body

Purpose

This configuration object allows you to alter the body of any SIP message for a matching session. You should only change the SIP message body under specific, required circumstances.

Syntax

```
config vsp policies policy default rule session-config
   inbound-header-settings altered-body
```

Properties

Property name	Description
admin [enabled | disabled]	When enabled, you can alter the body of any SIP message for a matching session. **Example: set admin disabled** The default setting is **enabled**.
altered-body	Alters the body of any SIP message for a matching session. **Example: set altered-body " (?ms) (.*) <address uri=\""sip: (.*) @ (.*) :5060; (.*) " " \1<address uri= :: sip: \2@10.1.1.1:5060; \4"**

Property name	Description
apply-to-methods	Specifies the message type to which the system applies message body changes. The system then changes the specified URI according to the settings of the header and destination properties of this object. When you modify this value, the system overwrites the current setting with only the message types you specify. To enter multiple types, enter them separated by a plus sign (+) with no spaces.The following are valid values: -INVITE -REFER -MESSAGE -INFO -OPTIONS -REGISTER -SUBSCRIBE -NOTIFY -PUBLISH -ACK -BYE -CANCEL -PRACK -UPDATE -SERVICE -PING **Example: set apply-to-methods notify+register** The default setting is **INVITE**.
apply-to-responses	Specifies whether to apply message body changes to just SIP requests or both requests and responses. If you enter a value of **yes**, you must include the response-code. The following are valid values: no—apply changes to requests only yes—apply to requests and responses **Example: set apply-to responses yes** The default setting is **no**.

altered-header

Purpose

The altered-header configuration object allows you to modify or create header values in calls matching this session configuration. You can create multiple header-altering configurations.

Syntax

```
config vsp policies session-policies policy rule session-config
   inbound-header-settings altered-header
```

Properties

Property name	Description
admin [enabled | disabled]	When enabled, you can alter header values in calls matching this session configuration. **Example: set admin disabled** The default setting is **enabled**.
source-header	Specifies the URI from which the system initially derives the data that is to be written to the destination header. **Example: set source-header P-Asserted-Identity**

Property name	Description
source-field	Specifies the portion of the URI that the system writes to the destination. The following are valid values: -user—The system writes the entire field to the destination. -host—The system writes the entire field to the destination. -selection—Specify the value within the URI to match on and the replacement text to write to the destination. -value—To write a specific string to the destination. In this case, the source-header field is ignored. **Example: set source-field user** The default setting is **““**.
destination	Specifies the header to be created or modified by the properties set in this object. The URI specified in this property is modified with the data from the **source-field** property. If the header doesn’t exist in the message, the AA-SBC creates it. The following are valid values: -To -From -Request **Example: set destination from**
destination-field	Specifies the field in the specified destination URI to overwrite. The following are valid values: -user -host -display -full (Overwrites entire selected destination URI.) **Example: set destination-field host** The default setting is **““**.

provisional-response

Purpose

The provisional-response object allows you to add any additional provisional responses you want sent along after the “100 Trying” response at the start of a normal INVITE dialog.

Syntax

```
config vsp policies session-policies policy default rule
   session-config provisional-response
```

Properties

Property name	Description
additional-response	Enter additional provisional responses you want sent with the INVITE dialog. **Example: set additional-response 101 “This is a test”**

codec-payload-type-bindings

Purpose

The codec-payload-type-bindings configures a binding between a codec name and a payload type. Without any codec-payload-type-bindings configured, the AA-SBC uses a default DTMF payload type of 101.

This configuration element is set when you want to change the default DTMF payload type offered by the AA-SBC. This property takes precedence over the default of 101. Codec-payload-type-bindings is used when the AA-SBC generates its own SDP for outgoing calls. The AA-SBC generates its own SDP for features like file-play or when the AA-SBC is in a Delayed-Offer/Early-Offer network.

Syntax

```
config vsp default-session-config in-media-normalization
   codec-payload-type-bindings
config vsp default-session-config out-media-normalization
   codec-payload-type-bindings
config vsp session-config-pool entry <name> in-media-normalization
   codec-payload-type-bindings
config vsp session-config-pool entry <name> out-media-normalization
   codec-payload-type-bindings
```

Properties

Property name	Description
binding	Bind a codec name to a particular payload type. **Example: set binding codec1**

63. Session configuration pool objects

Session configuration pool description

The session configuration pool is a mechanism for creating a session configuration that can be referenced through other objects. This allows you to create a specific configuration and re-use it for all applicable situations. See Chapter 21, “Dial plan objects”, for information on dial plans.

The objects available for configuration under a pool entry are the same session objects available for the default or pre-session configuration objects. See Chapter 62, “Session configuration objects”, for a complete description of each session configuration object.

Session config pool object summary

The following table lists and briefly describes the **session-config-pool** object. See the following chapter for other objects in the CLI hierarchy:

- Chapter 75, “Virtual system partition (VSP) objects”
- Chapter 62, “Session configuration objects”

Object name	Description
`session-config-pool`	Opens the session configuration pool configuration object.
`entry`	Creates an entry through which you define session configuration objects. See Chapter 62, “Session configuration objects” for a description of each session configuration object.

session-config-pool

Purpose

Opens the session configuration pool object through which you create re-usable session configurations. You can then reference these entries in dial plan, registration plan, calling groups, and server configurations, as well as the call-control action. Note that if you do create an individual session configuration under a dial-plan entry, that local session configuration takes precedence.

Syntax

```
config vsp session-config-pool
```

Properties

None

entry

Purpose

Creates an entry that can be referenced by a dial-plan or registration plan. Note that if you create an individual session configuration under a dial-plan entry, that local session configuration takes precedence. For details of the session configuration objects, see the Chapter 62, “Session configuration objects” descriptions.

Syntax

```
config vsp session-config-pool entry string
```

Properties

None

64. Settings objects

Settings description

The **settings** object controls advanced settings for a VSP. These are properties that you typically would not need to modify.

Settings object summary

The following table lists and briefly describes the **settings** object. See the following chapter for other objects in the CLI hierarchy:

- Chapter 75, “Virtual system partition (VSP) objects”

Object name	Description
settings	Sets a variety of VSP parameters. These are advanced settings that do not typically need modification.

settings

Purpose

Sets a variety of VSP parameters. These are advanced settings that do not typically need modification. You can configure basic VSP parameters using the vsp object.

Default plan types for SIP messages

Each SIP message type uses, by default, either a dial-plan or a registration-plan or no plan to determine how to handle matching messages. You can overwrite most default plan types for a given message using the **sip-message-plan** property. The settings that you establish with this property take precedence over the settings in the **apply-to-method** property of the dial- or registration-plan. Note that you cannot change the plan types for REGISTER or INVITE messages. The following table lists the default plan types

-

SIP message type	Plan type
INVITE	dial-plan
ACK	none
BYE	none
REGISTER	registration-plan
REFER	dial-plan
NOTIFY	registration-plan
OTHER	none
PRACK	none
CANCEL	none
SUBSCRIBE	registration-plan
OPTIONS	dial-plan
MESSAGE	dial-plan
INFO	dial-plan
PUBLISH	registration-plan
UPDATE	none
SERVICE	none

It is important to note that if you change the plan type for a message type, you must also update the applicable components of the plan through the **apply-to-methods** property. For example, NOTIFY uses a registration-plan by default. If you change it to use a dial-plan, you must modify the **apply-to-methods** property for normalization, source-normalization, arbiter, route, and/or source-route within the dial-plan to include NOTIFY as a method to which to apply the plan.

Configuring load balancing across OC client-to-LCS server tunnels

You can set up load balancing across tunnels using the **nnos-tunnel-creation** property. An OC client-to-LCS server tunnel actually consist of two separate connections that AA-SBC joins—a connection from the client to AA-SBC and a connection from AA-SBC to the server. This tunnel can be created either via the sip object or via the registration-plan. When using the SIP tunnel, you configure an IP interface with SIP, select a port matching the protocol the client is coming in on, and set the from- and to-server references.

However, if you want to load balance across tunnels, you must use tunnels created via the registration plan. To do so, ensure that you also have configured the following:

1. Enable the cluster-master, registration, and server-load master services.
2. Enable the **server-registration-balance** property in this settings object.
3. Set the **to-uri-match** property (route object) or **source-match** property (source-route object) in the registration plan object to a **type** of **local-port**. Enter the AA-SBC port number and IP address that clients come in on.
4. Set the **action** property in the registration plan **route** or **source-route** object to **tunnel**.
5. Create a pool of servers in the enterprise server object. These are the servers over which AA-SBC will balance calls.
6. Set the **peer** property in the registration plan **route** or **source-route** object to a **type** of **server**. For the **trunk** field, reference the server pool you created.
7. Set the **client-type** property in the registration plan **route** or **source-route** object to **windows-messenger** or **office-communicator**. This setting should be the same as the *fromServerReference* setting (for the appropriate transport) in the sip object.
8. Set the **rule** property in the registration plan arbiter object to **registration-balance**.

These steps enable load balancing across tunnels, using a weighted round robin algorithm. If you would like to control the order in which AA-SBC selects servers, do the following:

1. When configuring the server pool servers, configure a preference to influence the AA-SBC selection.

2. Add a second rule to the **rule** property in the registration plan arbiter object. Set the second rule to **most-preferred**.

Syntax

```
config vsp settings
```

Properties

Property name	Description
accounting-anonymous-match *regExp*	Specifies whether the system should modify the From header for accounting records. When accounting is enabled, if the From header of an INVITE message matches the specified regular expression, the system saves a different header URI value as the From header value in the Call Detail Record. This property works in conjunction with the **remote-party-id-accounting** property. If that property is **enabled** and there is a match to this property, the system uses the Remote-Party-ID as the From header; if **disabled**, the system uses the Contact header. Use this property in cases where some form of call blocking causes the From header to contain an anonymous or otherwise uninformative value. **Example: set accounting-anonymous-match *anonymous*** The default setting is **.*anonymous@localhost.***.
auto-server-failure-detection {enabled | disabled}	Enables or disables automatic detection of server failure and failover using SIP signaling messages. If this setting is **enabled**, the system clones each SIP message for retransmission, supporting registration and call failover from one server to another. In addition, the success or failure of a transmission is dispatched to the server pool to update state information at each server. This overhead does limit call scalability. However, if this setting is **disabled**, the system cannot perform failover. This setting is the master switch to enable this feature. You must also enable the feature for each intended server by setting the **failover-detection** property of the server object to **auto**, **ping**, or **register**. **Example: set auto-server-failure-detection enabled** The default setting is **disabled**.

Property name	Description
clear-binding-on-connection-broken {enabled | disabled}	Specifies whether the system deletes bindings in the location cache when a TCP or TLS connection is not in a connected state. When **enabled**, the system removes the bindings; when **disabled**, the bindings remain. **Example: set clear-binding-on-connection-broken disabled** The default setting is **enabled**.
connection-timeout *seconds*	Sets the number of seconds that a SIP TLS or TCP connection can remain idle before the system closes it. Setting the value too low can cause the system to have to re-establish the connection frequently. A value of 0 disables the timeout function. **Example: set connection-timeout 300** Enter a value from 0 to 86,400 seconds. The default setting is **0** (disabled).
database-write {enabled | disabled}	Sets whether data is written to the main database. If you disable this feature, the system writes nothing further to the database but previous records remain until they are cleaned out by the maintenance process. See the database object in Chapter 39, “Master services objects” for more information on maintenance. **Example: set database-write disabled** The default setting is **enabled**.

Property name	Description
filter-mcs-authint-to-auth {enabled | disabled}	Specifies whether the system changes the Quality of Protection (QOP) value, reducing the option to only auth (authentication). This value is used between Nortel clients and servers to define the level of authentication—auth and/or authentication with integrity (authint). When **enabled**, the system replaces the authint value in the header with the value auth. If the header contained both auth and authint, the system simply removes authint. This allows the system to rewrite the message, which is necessary for anchoring calls. When **disabled**, the header is left unchanged, but the system cannot anchor calls. **Example: set filter-mcs-authint-to-auth disabled** The default setting is **enabled**.
filter-mcs-force-IM-decrypt {enabled | disabled}	Specifies whether the filter function modifies the Nortel header that specifies encryption. (Nortel encrypts all SIP payload by default.) When **enabled**, the system changes the header so that it informs the server not to re-encrypt the message when sending it to the far end. When **disabled**, the message is re-encrypted. However, the system cannot perform IM filtering on encrypted messages, so if you set it to disabled, IM filtering is disabled as a result. **Example: set filter-mcs-force-IM-decrypt disabled** The default setting is **enabled**.

Property name	Description
filter-mcs-independent-header-schemes {enabled | disabled}	Determines how the system modifies the To and From headers of the SIP message to allow compatibility with MCS (which does not currently support SIPS). As a result, the system is required to convert the “sips:” portion of header to “sip:” before delivering to MCS and then must restore the scheme for messages (of all types) returned to SIP clients. If this property is **enabled**, the system restores the scheme part of the URI in both the To: and From: headers from the state saved in the header parameters in the message itself. If **disabled**, the system restores the scheme based on the type of tunnel connecting it to the client (TLS or non-TLS). **Example: set filter-mcs-independent-header-schemes enabled** The default setting is **disabled**.
filter-mcs-rewrite-ping-contact-hdr {enabled | disabled}	Specifies whether the Nortel MCS filter in the system rewrites the PING contact header. When **enabled**, the system changes the contact header to report the client-visible public IP address instead of the client-side address of the system. This is useful if the client is behind a far-end NAT. **Example: set filter-mcs-rewrite-ping-contact-hdr enabled** The default setting is **disabled**.
filter-mcs-site-failover-threshold *seconds*	Sets the amount of time the system waits for a response to a SIP message before conducting a site failover. This property is used in network configurations that use redundant sites. When the system expects some activity from a site (for example, a response to a REGISTER), it allows this interval to pass before performing a DNS lookup to determine the new, redundant site. **Example: set filter-mcs-site-failover-threshold 300** The default setting is **240** seconds (four minutes).

Property name	Description
filter-mcs-suppress-100rel {enabled | disabled}	Specifies whether the system modifies the Nortel Supported Header field (requesting acknowledgement) in the SIP request. When **enabled**, the system removes the field, causing the far-end agent not to request a reliable response. When **disabled**, the system leaves the header field untouched. **Example: set filter-mcs-suppress-100rel disabled** The default setting is **enabled**.
filter-lcs-input-remove-user-params {enabled | disabled}	Enables or disables the LCS filter that strips User parameters from INVITES. By default (**enabled**), the system applies the LCS filter and strips the parameters. When **disabled**, the system leaves the user parameters in tact. **Example: set filter-lcs-input-remove-user-params disabled** The default setting is **enabled**.
filter-lcs-input-remove-record-route-hdrs {enabled | disabled}	Specifies whether the system strips the Record-Route header from SIP messages. This setting is only applicable in configurations that are using messaging client tunnels (the system acts as a proxy between a messaging client and its native server). When **enabled**, the system strips the headers, when **disabled**, it leaves the message headers intact. Leave this setting at the default, enabled, for compatibility with LCS 2005. **Example: filter-lcs-input-remove-record-route-hdrs disabled** The default setting is **enabled**.

Property name	Description
filter-lcs-input-remove-bye-ack-params {enabled | disabled}	Enables or disables the LCS filter that strips the parameters of the REQUEST URI from any BYE or ACK messages. By default (**enabled**), the system applies the LCS filter and strips the parameters. When **disabled**, the system leaves the user parameters in tact. Set this to **disabled** for use with OCS 2007. **Example: filter-lcs-input-remove-bye-ack-params disabled** The default setting is **enabled**.
ignore-contact-on-ack {enable | disable}	Specifies how the system updates call leg remote contact information. When **enabled**, the system does not update the call leg remote contact information from the contact header of the SIP message ACK. If **disabled**, the system uses information in the contact header of an ACK message to update the call leg remote contact field. Further, it uses that contact information when forwarding the future request message. **Example: set ignore-contact-on-ack enabled** The default setting is **disabled**.

Property name	Description
location-cache-write-thru {enabled | disabled}	Specifies when to flush modifications into the location database. During the SIP process, location data is changed incrementally with each REGISTER and INVITE request. These incremental changes are saved in a location cache entry attached to the SIP message. If a new message has to access or modify the location data for the same AOR, the system transfers the cache copy, along with any previously modified data, to that new message. The cache copy persists from one message to another until a cache-holder message is destroyed. At that point, the cache copy is flushed into the location database. When **enabled**, any modification to the location data is immediately flushed into the location database. When **disabled**, the system waits until the SIP message is destroyed. **Example: set location-cache-write-thru enabled** The default setting is **disabled**.
sip-message-plan *sipMsgTypes* {none | dial-plan | registration-plan}	Changes or assigns a plan type that the system uses for different SIP message types. These settings take precedence over the apply-to-method settings of the dial-plan or registration-plan. You can assign a plan type or no plan (**none**) to each message type. REGISTER requests always use the registration-plan and INVITEs always use the dial-plan; you cannot change these plan types. See Default plan types for SIP messages for a more complete explanation and important note. **Example: set sip-message-plan NOTIFY dial-plan** There is no default setting. However, if you enter a message type, the default plan type is **dial-plan**.

Property name	Description
request-line-routing {enabled | disabled}	Determines whether message forwarding is done based on the To: header or the Request URI. If **enabled**, the system makes its forwarding decision based on the Request URI. If **disabled**, the system makes its forwarding decision based on the To: header. **Example: set request-line-routing disabled** The default setting is **enabled**.
calling-group-routing {enabled | disabled}	Specifies whether to include calling-groups in the dial plan search criteria when determining how to forward a call. If **enabled**, the system applies the calling-group group match criteria. Incoming calls are checked to see if the IP address matches a calling group. If there is an IP match, the call is forwarded according to the routes configured under that calling group.If **disabled**, the system ignores any routes in the calling-groups configuration. Only the global dial-plan routes are searched when trying to route a call from a calling group. **Example: set calling-group-routing enabled** The default setting is **disabled**.
resolve-routing-through-server-domain {enabled | disabled}	For Technical Support use only. Do not enable this property without explicit instructions to do so.
max-options-retransmissions *attempts*	Specifies the maximum number of times the system attempts to retransmit SIP OPTIONS messages at the transaction level. By default the system only transmits OPTIONS messages once, so that it can quickly detect failure. Set the value to a higher number of attempts to allow a longer time for successful OPTIONS response. Use the session config sip-settings **max-retransmission** property to control retransmission of other message types. **Example: set max-options-retransmissions 3** Enter a value from 1 to 32; the default setting is **1** attempt.

Property name	Description
max-udp-outbound-log *integer*	Sets the total number of log entries allowed. The system creates logs when you execute the sip **udp-log-on** action. Use this option for debugging only. **Example: set max-udp-outbound-log 100** Enter a value from 0 to 30,000. The default setting is **30**.
options-forward {enabled | disabled}	Specifies whether the system, when acting as a proxy, forwards OPTIONS messages to the UAS. When **enabled**, the system forwards the messages to the provider and returns the response to the SIP client (UAC). When disabled, the system does not forward OPTIONS messages. **Example: set options-forward enabled** The default setting is **disabled**.

Property name	Description
out-of-context-message-action {allow | discard | refuse [*resultCode*] [*resultString*]}	Specifies the action to take upon receipt of a SIP message that is not affiliated with a current system session. An out of context message is one that arrives at the system with a Call-ID different from that of any currently active session. If the arriving message is a request, and is of a method type that could start a new session (for example, INVITE, and in some cases MESSAGE, NOTIFY, SUBSCRIBE) then it is permitted. If the arriving message is a response message, or a request of a method type that can only occur within an already established session (for example, BYE, CANCEL, INFO), then it is labeled “out of context” and the prescribed action is performed. Action options are: • **allow**—Allow the message to be processed, and possibly forwarded, by the system SIP stack. • **discard**—discard out-of-context messages without a notification. • **refuse**—discard the packet but send a response to indicate having done so. The response includes an error code and an optional description. **Example: set out-of-context-message-action refuse 481 “bad callID”** The default setting is **discard**. If you select refuse, the default result code is 400; you can enter any value between 400 and 699. There is no default result string.
out-of-context-message-media-cleanup {enabled | disabled}	Specifies whether the system should tear down a media session that corresponds with an out of context message. When **enabled**, the system removes the session. See the **out-of-context-message-action** property for a description of that message type. **Example: set out-of-context-message-media-cleanup disabled** The default setting is **enabled**.

Settings objects

Property name	Description
preserve-3xx-contact {enabled | disabled}	Specifies whether the NAT filter service on the system should modify the contact header in a 3*xx* response message when routing an outbound call. Normally (when **disabled**), the system resets the contact header to its own local IP address. When **enabled**, the NAT filter service makes no modification to the header. Note that the preserve-3xx-contact in the sip-settings object controls session-based change for the SIP stack. **Example: set preserve-3xx-contact enabled** The default setting is **disabled**.
prune-associations {enabled | disabled}	Specifies whether the system should remove inactive associations from the location database to reclaim memory. When **enabled**, the system removes inactive sessions with the frequency defined in the **pruning-interval** property. **Example: set prune-associations disabled** The default setting is **enabled**.
pruning-interval *seconds*	Specifies the frequency with which the system attempts to reclaim inactive associations. **Example: set pruning-interval 7200** Enter a value from 1 to 360,000; the default setting is **3600** (one hour).
read-header-max *characters*	Sets the maximum character length of the SIP header. This property provides buffer overflow control. If the maximum character length is exceeded, the message is discarded. Note that if the message arrived on a TCP or TLS socket (as opposed to UDP) the connection is also closed when the message is discarded. **Example: set read-header-max 1028** Enter a value between 64 and 65535. The default setting is **4095** characters.

Property name	Description
read-line-max *characters*	Sets the maximum character length for lines in the SIP message. This property provides buffer overflow control. If the maximum character length is exceeded, the entire message is discarded. If a message line has one or more continuation lines, the lengths of all these lines are added together. Note that if the message arrived on a TCP or TLS socket (as opposed to UDP) the connection is also closed when the message is discarded. **Example: set read-line-max 800** Enter a value between 64 and 4095. The default setting is **2047** characters.
read-message-max *characters*	Sets the maximum length of an entire message, including SIP header and SDP (Session Description Protocol) or other message body. This property provides buffer overflow control. If the maximum message length is exceeded, the message is discarded. Note that if the message arrived on a TCP or TLS socket (as opposed to UDP) the connection is also closed when the message is discarded. **Example: set read-message-max 65535** Enter a value between 64 and 65535. The default setting is **32768** characters.
malformed-message-silent-drop {enabled | disabled}	Specifies whether the system should drop malformed packets that arrive on a SIP port but do not pass SIP parsing. When **enabled**, the system drops messages identified by the kernel as malformed without sending them to the SIP process. When **disabled**, the malformed messages are sent to the SIP process, which may log a message, record the malformed message to the database for DOS pattern detection, or both, depending on other configuration settings. **Example: set malformed-message-silent-drop enabled** The default setting is **disabled**.

Property name	Description
splittable-headers *header*	Specifies header handling when there are multiple instances of the same header type in a message. If you specify a header type with this property and multiple instances of that type are encountered in the message, AA-SBC splits each instance onto a separate line. Multiple instances of a header type that is not specified here results in AA-SBC combining all instances into a single, comma separated line. You can specify as many header types as necessary by re-executing the command. Note that the following headers are always split on to multiple lines, regardless of the configuration: • Contact • Record-Route • Route • Via The following headers are always combined, regardless of the configuration: • Allow • Allow-Events **Example: set splittable-headers Diversion** The default setting is **disabled**.

Property name	Description
server-redirect-service {enabled | disabled}	Specifies whether the system saves the per-AOR redirect state of a server to its location cache. The redirect server is an alternative server listed in the Contact field of a SIP response header. If this property is **enabled**, when the system delegates a REGISTER request to a server and gets a redirect response (301/302), it performs as follows: 1. Saves the per-AOR redirect state of the server to the location cache for the AOR in concern. 2. Changes the response to a 200OK and forwards it to the destination. 65. Changes the expiration time for the register to a brief interval, causing the registering phone to reregister. When the REGISTER arrives, the system does a location cache lookup which reports the redirect state, and forwards the REGISTER to the new server. If the system receives a call (instead of a REGISTER request) and either the To or From fields contain an AOR where the server known state is redirect, the system forwards the call to an alternative server if both of the following conditions are met: 1. the system has previously redirected the AOR associated with the From or To fields. 66. A dial-plan lookup determines that the next-hop is a server whose state for the AOR is set to redirect. If set to **disabled**, the system does not save the state information. **Example: set server-redirect-service enabled** The default setting is **disabled**.

Property name	**Description**
server-registration-balance {enabled | disabled}	Sets registration load balancing on a global level. When **enabled**, all configured servers will participate in load-balancing of REGISTER requests. Balancing is done in proportion to the maximum number of requests allowed on each server (set by the server-pool-admission-control **max-number-of-registrations** property). Note that once a REGISTER has been forwarded to a particular server, all future messages intended for that AOR will be forwarded to the correct server. Note that the registration-plan arbiter object, if it contains a **registration-balance rule**, takes precedence over this setting. **Example: set server-registration-balance enabled** The default setting is **disabled**.
nnos-tunnel-creation {interface | registration-plan}	Specifies whether OC client-to-LCS server tunnels are configured via the sip interface or derived from the registration-plan. Set the property to **registration-plan** if you want to load balance across tunnels. See Configuring load balancing across OC client-to-LCS server tunnels for a complete description of the configuration requirements to complete tunnel load balancing. **Example: set nnos-tunnel-creation registration-plan** The default setting is **interface**.

Property name	Description
skip-via-transport-check {enabled | disabled}	Specifies whether to ignore a mismatch in the Via header. Normally, the transport type (TCP, UDP, TLS) in the top Via header must match the transport protocol of the SIP message the system received. Use this property in cases where the client does not follow that structure. For example, the client message may have TCP in the top Via header, but actually the message was received from a TLS connection. If set to **disabled**, when the system finds this kind of mismatch, it discards the message. If this property is **enabled**, the system does not perform the check, resulting in allowing the mismatch. **Example: set skip-via-transport-check enabled** The default setting is **disabled**.
socket-receive-buffer-size *megabytes*	Specifies the kernel socket buffer size for the SIP stack—for receiving SIP messages. If the system reaches the buffer size, it informs the sender so that the sender can slow transmission. **Example: set socket-receive-buffer-size 2** Enter a value from 1 to 64; the default setting is **1** megabyte.

Property name	Description
sockets-idle-max *seconds*	Sets the maximum number of seconds that a TCP connection can remain idle before closing the connection. Every 10 seconds, the system scans all the open TCP connections. If idle TCP connections are found, and if those connections have been idle for at least the number of seconds specified by the **sockets-idle-max** parameter, the idle TCP connections are closed. If the value is set to 0 (the default), idle sockets can remain open indefinitely, as long as their resources are not needed for a new connection. **Example: set sockets-idle-max 15** Enter a value between 0 and 65535. The default setting is **0** seconds.
sockets-idle-min *seconds*	Sets the minimum number of seconds that can transpire before a TCP connection is officially declared idle and available for a new TCP connection (socket). If a new TCP connection is opened, and if the total current open TCP connection count exceeds the limit specified by the **sockets-per-box-max** parameter setting, the system attempts to find an open connection that has been idle for at least the number of seconds specified by the **sockets-idle-min** property. If an idle TCP connection is found, the connection is closed and available for a new TCP connection. If there are no TCP connections that have been idle for the specified number of seconds, none are closed. If the **sockets-per-box** limit is exceeded by more than 10, new connections are refused until some of the existing sessions have closed or have been declared idle. **Example: set sockets-idle-min 10** Enter a value between 1 and 65535. The default setting is **5** seconds.

Property name	Description
sockets-initial-message-timeout *seconds*	Sets the number of seconds the system waits for a valid SIP message, once a TCP or TLS connection is established. If the timer expires, the system disconnects the call. **Example: set sockets-initial-message-timeout 3** Enter a value between 0 and 600. The default setting is **5** seconds.
sockets-per-box-max *integer*	Sets the maximum number of open TCP sockets (connections) to allow on this VSP. If the maximum number of connections is reached, the system first attempts to find idle or invalid connections to shutdown before refusing to accept new connections. **Example: set socket-per-box-max 1000** Enter a value between 1 and 100,00. The default setting is **1024** sockets.
sockets-per-peer-max *integer*	Sets the maximum number of open TCP sockets (connections) one remote address (a VSP peer) can have open with the system at one time. If the maximum number of connections is reached, the system first attempts to find idle or invalid connections to shutdown before refusing to accept new connections. Note that if the per-box and per-peer maximums are the same, a peer could potentially control all connections to the box, for example, in the case of a DOS attack. **Example: set socket-per-peer-max 1000** Enter a value between 1 and 100,00. The default setting is **256** sockets.

Property name	Description
stack-message-queue-max *messages*	Specifies the maximum number of messages that the system can queue in the SIP worker threads. When the processing queue length reaches the value set with this property, the system stops reading new messages from the network, and instead works on clearing out the backlog. (These messages are not deleted but saved until read.) It remains in this mode until the queue length gets down to **stack-message-queue-min**. At that point, it resumes reading new messages from the network. **Example: set stack-message-queue-max 2048** Enter a value from 256 to 65,536; the default setting is **8192**.
stack-message-queue-min *messages*	Specifies the number of messages the message queue must be reduced to before the system can begin queuing new messages. See the description for **stack-message-queue-max** for a complete description. **Example: set stack-message-queue-min 256** Enter a value from 0 to 65,536; the default setting is **7168**.

Property name	Description
stack-message-queue-reg-clip-threshold *messages*	Specifies the maximum number of messages allowed on the SIP stack transport processing queue before the system begins discarding incoming REGISTER messages. As long as the queue length is over this number, the system continues to discard new REGISTER requests, but still queues other traffic for processing. The system reads REGISTERs from the network, but then discards them because the system is too busy to process them. Note that if the value set for this property is higher than the value of **stack-message-queue-max**, the system will never reach the clipping threshold. **Example: set stack-message-queue-reg-clip-threshold 100** Enter a value from 0 to 65,536; the default setting is **0** (disabled).
supported-extensions *string*	Specifies the name of any extensions to SIP that the system should allow for processing. Some SIP REQUEST messages contain a field indicating that the endpoint must support particular extensions. By default, the system rejects those messages. To enable passage of those messages, enter the extension name(s) in this property. The system does not provide support for those extensions, but will recognize them and will not reject the message. **Example: set supported-extensions ABCco-SIPvendor.extension** There is no default setting. The default action is for the system to reject messages containing supported extension requirements.

Property name	Description
translate-sips-scheme {enabled | disabled}	Specifies whether to change the sips: portion of the header to sip: in the REQUEST, TO, or FROM URI and the Contact and Via headers. If enabled, the system changes the secure SIP header (sips:) to plain sip. Use this in cases where the destination server does not support secure SIP (for example, most phones do not support sips). **Example: set translate-sips-scheme enabled** The default setting is **disabled**.
tunnel-policy {enabled | disabled}	Specifies whether matching policy modifies the SIP message when messaging client tunnels are configured. By default, policy does not change messages in this tunnel environment. When enabled, changes dictated by matching policy will edit the SIP message. See Configuring Messaging Client Tunnels in the sip object for more information. **Example: set tunnel-policy enabled** The default setting is **disabled**.
udp-ignore-content-length {false | true}	Specifies whether the system ignores the content length field of the SIP header for packets coming in over UDP. The setting only applies when the content length field shows a value greater than the actual content length of the SIP message body. When set to **false**, the system does not ignore the field, and therefore discards any packet arriving with an actual content length that does not match the value in the content length field. When set to **true**, the system forwards the packet. Set this field to **true** in a case where a SIP proxy or user agent incorrectly calculates and rewrites the content length field. This may happen, for example, when a NAT device is in use. **Example: set udp-ignore-content-length true** The default setting is **false**.

Property name	Description
udp-tunnel-reclaim {enabled | disabled}	Specifies whether the system should scan for, and tear down, inactive UDP-to-UDP tunnels. When **enabled**, the system removes tunnels that are determined inactive by expiration of the time set with the static-stack-settings **max-udp-tunnel-inactivity** property. Use the **udp-tunnel-reclaim-scan-interval** property to set the frequency with which the system checks for idle tunnels. **Example: set udp-tunnel-reclaim disabled** The default setting is **enabled**.
udp-tunnel-reclaim-scan-interval *seconds*	Sets the frequency with which the system checks for idle tunnels. The static-stack-settings **max-udp-tunnel-inactivity** property defines the number of seconds that a tunnel can remain inactive before being deemed idle. When the udp-tunnel-reclaim property is **enabled**, the system tears down the idle tunnels found with each scan. **Example: set udp-tunnel-reclaim-scan-interval 120** Enter a value between 60 and 360,000. The default setting is **600** seconds.

Property name	Description
cisco-79xx-auto-ack {enabled | disabled}	Specifies whether the system requires an ACK response from an INVITE challenge. Typically, when a system receives an INVITE that it must challenge, it sends a 401 challenge and awaits an ACK in response. The phone can then resend the INVITE with the appropriate Auth header information and the call can proceed. This property is for use with Cisco phones models 7940 and 7960. These phones do not send an ACK, and instead just resend the INVITE. The system responds to the new INVITE with a 500 Server Internal Error. Set this to **enabled** if you have a Cisco 79xx model phone to signal the system to accept the new INVITE without the ACK. **Example: set cisco-79xx-auto-ack enabled** The default setting is **disabled**.
lowercase-sip-addrs {true | false}	Specifies whether the system changes case for SIP addresses before storing them in the database. By default (**true**), the system changes to lowercases all addresses before storage. However, in some instances of mixed case addresses, association lookups will fail if the address has been stored as lower case. Set this property to **false** to store addresses in the database as received—either lower or mixed case. **Example: set lowercase-sip-addrs false** The default setting is **true**.
sip-process-auto-restart {enabled | disabled}	Specifies the action the system should take if it detects a fatal error (e.g., deadlock) in the SIP process. If **enabled**, the system will either restart the process or, if configured with vrrp, failover to the backup box. If **disabled**, the process remains down and the system sends a warning message to the error log. **Example: set sip-process-auto-restart disabled** The default setting is **enabled**.

Property name	Description
prescan-media-types {enabled | disabled}	Sets whether the system prescans SIP messages for media descriptions. When **enabled**, the system preprocesses messages for CODEC-based routing. When **disabled**, it forwards messages through. You must enable this property if you have set a dial plan or policy condition-list to match on the sip-message-condition **media-types** attribute. Otherwise, this property should be disabled to avoid unnecessary processing overhead. **Example: set prescan-media-types enabled** The default setting is **disabled**.
backup-server *ipAddress* {any | UDP | TCP | TLS}	Assigns a backup server for use with the system hitless upgrade feature. There is a window of time between the beginning of the upgrade process and the point at which the system can no longer accept calls. During this time, the system redirects any calls that come in to it to the backup server specified. Therefore, a call is not started and then stranded. **Example: set backup-server 10.10.10.1 tls** There is no default setting.
register-retransmit-detection {enabled | disabled}	Specifies whether the system forwards request messages that were resent because the response to the original request message was dropped. When **enabled**, the system does not retransmit a client resent request message. When **disabled**, the system does resend the request. **Example: set register-retransmit-detection disabled** The default setting is **enabled**.

Property name	Description
remote-party-id-accounting {enabled | disabled}	Specifies the header value that the system should replace the From header with in the Call Detail Record in certain configurations. If the From header of an INVITE message matches the regular expression specified in the **accounting-anonymous-match** property, and this property is **enabled**, the system uses the Remote-Party-ID as the From header. If **disabled**, the system uses the Contact header as the From header. **Example: set remote-party-id-accounting enabled** The default setting is **disabled**.
apply-to-methods *messageType*	Specifies the message type(s) that the system allows to be processed by an external policy server. When this message type is received by the system, if there is an external policy server configured, the system forwards a WSDL request to that server for the specific policy to apply to the session. **Example: set apply-to-methods INVITE+REFER** The default setting is **INVITE**.
stack-socket-threads-max {automatic | *threads*}	*Secondary property*. Sets the number of SIP stack processing threads that should be used for TLS processing. A greater number of threads speeds up TLS connection establishment. If you are not using TLS, set the threads to 1. If you are using TLS, set the value to 4. **Example: set stack-socket-threads-max 4** The default value for this property is **automatic**. See Using automatic values for more information.

Property name	**Description**
stack-socket-event-threads-max {automatic | *threads*}	*Secondary property*. Sets the number of threads dedicated to servicing “RX available” events on the SIP sockets. These threads read the data from the sockets (in the case of TLS, this data is already decrypted), complete parsing of the data into complete SIP messages, and queue these messages for the worker threads to process. In setting this value, it is best to set it lower than the value set for the static-stack-settings **stack-worker-threads-max** property. **Example: set stack-socket-event- threads-max 40** The default value for this property is **automatic**. See Using automatic values for more information.
sip-socket-event-queue-max *events*	*Secondary property*. Sets the maximum number of socket events that can be queued for a single socket at any given time. Do not change this property unless instructed to do so by Technical Support. **Example: set sip-socket-event-queue-max 4** The default setting is **4**.

Property name	**Description**
local-directory-based-user-services {enabled *msgType* | disabled}	*Secondary property.* Toggles whether to perform policy and user services on this VSP (whether to use policy to secure and control SIP traffic). You should only set this to **enabled** if you need to perform directory-based user services (vsp/ enterprise/directory) for your SIP traffic. If enabled, you can also select which SIP messages trigger the creation of associations and user group lookups. Note that if this property is **enabled**, you must set the directory services for the cluster. Also, if you enable this *without* configuring the server **directory** property (to assign a directory to a server), you must configure the server **domain** name in order to match user SIP addresses to the appropriate server (by use of the domain). **Example: set local-directory-based-user-services enabled invite+refer+register** The default setting is **disabled**. If enabled, the default triggers are REGISTER and INVITE messages.
session-list-enable {enabled | disabled}	*Secondary property.* Specifies whether or not to maintain a session list (a list of all sessions to/ from a given URI). This property must be **enabled** for tone insertion to work. If the system receives a CSTA signaling message containing DTMF digits, it replaces1 second of audio with the DTMF RTP packets (normally 50 RTP packets) in the RTP media stream. In that way, both CSTA and non-CSTA endpoints can incorporate the tones. **Example: set session-list-enable enabled** The default setting is **disabled**.

Property name	Description
sip-process-overload-restart {enabled | disabled}	*Secondary property.* Restarts the SIP process if the system detects an overload of the SIP stack. This causes a crash of the SIP process and creates a crash file for debugging purposes. Do not enable this property unless instructed to do so by Technical Support. **Example: set sip-process-overload-restart enabled** The default setting is **disabled**.
check-sip-sockets {enabled | disabled}	*Secondary property*. Determines whether the system should check the state of all the server sockets periodically and issue a RX event, if necessary. Enable this to run a troubleshooting check if you determine that packets are not coming through. When **enabled**, the system runs a check twice a second. **Example: set check-sip-sockets enabled** The default setting is **disabled**.
timeout *milliseconds*	*Secondary property.* Specifies the maximum amount of time an external server has to respond to a policy request. If the timer expires, the system aborts the request. **Example: set timeout 20000** Enter a value between 100 and 30,000; the default setting is **30,000**.
max-queued-messages *messages*	*Secondary property.* Specifies the maximum number of concurrent external policy requests allowed. When this value has been reached, the system cannot consult an external policy server to forward the SIP request. Instead, it forwards/ processes requests using only the internal system policy. This condition remains until the queue drops below this limit. **Example: set max-queued-messages 150** The default value is **0**, unlimited.

Property name	Description
lnp-tracking {enabled | disabled}	*Secondary property.* Provides a customer-specific application implementation and is not otherwise applicable. The default setting is **disabled**.
lnp-timer *minutes*	*Secondary property.* Provides a customer-specific application implementation and is not otherwise applicable. The default setting is **15** minutes.
lnp-removal *minutes*	*Secondary property.* Provides a customer-specific application implementation and is not otherwise applicable. The default setting is **14400** minutes.
send-trying-before-stack {enabled | disabled}	*Secondary property.* Determines at what point in processing AA-SBC sends a "100 Trying" response to an INVITE. When this property is **enabled**, the system sends the response when low-level processing is occurring, before SIP stack processing begins. When **disabled**, the system sends a response when the SIP stack processes the INVITE. **Example: set send-trying-before-stack disabled** The default setting is **enabled**.
track-sip-responses {enabled | disabled}	*Secondary property.* Specifies whether AA-SBC tracks the responses to SIP REGISTER and INVITE messages. If **enabled**, the **show sip-register-responses** and **show sip-invite-response** status providers include data indicating the type and number of responses sent and received (e.g., the number of 200 OKs, 503s, etc.). **Example: set track-sip-response disabled** The default setting is **disabled**.

Property name	Description
check-content-headers-method	*This is a secondary property*. Checks the incoming SIP message Content-Length and Content-Type headers to make sure they exist when the message body is not empty. This property determines which method type to do this check. By default, all the methods are subject to check. You can specify one of the following: INVITE, REFER, MESSAGE, INFO, OPTIONS, REGISTER, SUBSCRIBE, NOTIFY, PUBLISH, ACK, BYE, CANCEL, PRACK, UPDATE, SERVICE, or PING. **Example: set check-content-headers-method refer**
check-content-headers-level	*This is a secondary property.* Checks the incoming SIP message Content-Length and Content-Type headers to make sure they exist when the message body is not empty. This property determines which header to check. The following are valid values: -none—Do not check headers -content-type—Check Content-Type header -Content-Type+Content-Length—Check both Content-Type and Content-Length headers. **Example: set check-content-headers-level none** **The default setting is content-type.**

65. Session Initiation Protocol objects

Session Initiation Protocol description

The *Session Initiation Protocol (SIP)*, described by RFC 3261, is the Internet protocol that establishes, modifies, and terminates conferencing and telephony sessions over an IP-based network using text-based messages. SIP is a major protocol in real-time collaboration networks.

You enable and configure SIP on Ethernet and VLAN interfaces. To configure load-balancing of SIP processing, see Configuring head-end and backing interfaces.

Network Address Translation

Network Address Translation (NAT) takes the internal IP addresses from the private network and maps them to global public IP addresses for recipients on the public Internet. When an internal IP *address:port* (source address) is mapped to an external IP *address:port* (destination address), recipients can route traffic back to the originating IP address and port. NAT protects the private IP addresses from being exposed to clients on the public Internet.

On AA-SBC

AA-SBC uses NAT to ensure that SIP phone calls from internal clients on the private network can traverse enterprise firewalls en route to external clients on the public Internet. NAT operates on the two components that comprise a SIP phone call—the SIP signaling stream that sets up the phone call, and the media stream that carries RTP packets between the SIP clients. This includes:

- Re-writing IP address and TCP/UDP port information embedded in SIP/SDP messages as necessary to ensure address continuity
- Opening and closing internal media ports (“pinholes”) and controlling NAT bindings dynamically, in perfect synchronization with SIP signaling state to enable secure transit of SIP-associated media streams.

SIP object summary

The following table lists and briefly describes the **sip** objects. See the following chapters for other objects in the CLI hierarchy:

- Chapter 14, "Cluster, box, and interface objects"
- Chapter 35, "IP objects"
- Chapter 77, "VLAN objects"
- Chapter 78, "VRRP objects"

Object name	Description
`sip`	Configures SIP on AA-SBC interfaces.
`load-balancing`	Configures load-balancing method and backing interfaces.

sip

Purpose

Configures the Session Initiation Protocol (SIP) on an Ethernet or VLAN interface.

Syntax

```
config cluster box number interface ethX ip name sip
config cluster vrrp vinterface vxID ip name sip
config cluster box number interface ethX vlan number ip name sip
config box interface ethX ip name sip
config box interface ethX vlan number ip name sip
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables SIP on this IP interface. **Example: set admin enabled** The default setting is **disabled.**
nat-translation {enabled | disabled}	Enables or disables NAT translation on this interface. See Network Address Translation for more information. **Example: set nat-translation enabled** The default setting is **disabled**.
nat-add-received-from {enabled | disabled}	Sets whether the system modifies the FROM header on a NAT-translated session. If **enabled** (and far-side NAT translation is enabled), when the system transmits an INVITE, it adds a “received-from” parameter to the From: header. The property includes the public IP address on which the original REGISTER was received. **Example: set nat-add-received-from enabled** The default setting is **disabled**.
udp-port *portNumber* [*fromServerReference*] [*toServerReference*]	Sets the User Datagram Protocol (UDP) port number to use when listening for SIP messages. The known UDP port number for SIP is 5060. Optionally, you can enter a reference to a source and destination server to enable tunneling for Nortel clients. Setting the server “tells” the system that all traffic on this port is between those server types, enabling the system to filter based on that information. Use quotation marks to enter the reference. See Configuring Messaging Client Tunnels for more information. **Example: set udp-port 5060 “vsp\enterprise\servers\ sip-host nortel-client” “vsp\enterprise\servers\mcs mcs-server”** The default setting is **5060**.

Property name	Description
tcp-port *portNumber* [*fromServerReference*] [*toServerReference*]	Sets the Transmission Control Protocol (TCP) port number to use when listening for SIP messages. The known TCP port number for SIP is 5060. Optionally, you can enter a reference to a source and destination server to enable tunneling for Windows Messenger clients. Setting the server “tells” the system that all traffic on this port is between those server types, enabling the system to filter based on that information. Use quotation marks to enter the reference. See Configuring Messaging Client Tunnels for more information. **Example: set tcp-port 5060 “vsp\enterprise\servers\ sip-host WMsgr” “vsp\enterprise\servers\lcs lcs-server”** The default setting is **5060**.
tls-port *portNumber* [*fromServerReference*] [*toServerReference*]	Sets the TLS port number to use when listening for SIP messages. The known TLS port number for SIP is 5061. Optionally, you can enter a reference to a source and destination server to enable tunneling for Windows Messenger or Nortel clients. Setting the server “tells” the system that all traffic on this port is between those server types, enabling the system to filter based on that information. Use quotation marks to enter the reference. See Configuring Messaging Client Tunnels for more information. **Example: set tls-port 5061 “vsp\enterprise\servers\ sip-host WMsgr” “vsp\enterprise\servers\lcs lcs-secure”** The default setting is **5061**.
certificate *certificateReference*	Assigns the certificate that must be presented to participate in SIP exchanges. Enter a reference to a previously configured certificate. **Example: set certificate vsp tls certificate nnos-e.companyA.com** There is no default setting.

load-balancing

Purpose

Configures load balancing backing interfaces and distribution method. Note that typically load-balancing is configured on VRRP interfaces to create the redundancy. You must configure the load-balancing master service for load balancing to be enabled.

To load balance across tunnels, see Configuring load balancing across OC client-to-LCS server tunnels for complete configuration instructions.

Configuring head-end and backing interfaces

Load balancing of SIP processing across interfaces requires both headend and backing interfaces. The *head-end* interface is the central distribution point. It does not do any SIP processing, it only forwards the calls to its configured backing interfaces. When you configure a SIP phone, you would configure it to point to the head-end interface. The *backing interface* is the location at which AA-SBC terminates TCP and TLS connections (and where UDP transport messages arrive) and handles SIP processing. AA-SBC uses load-balancing (the method is configured with the **load-balancing mode** property) to distribute messages across the configured backing interfaces.

To configure an IP interface as a head-end interface, navigate to the backing interface(s) it will use and configure the **load-balancing** object **head-end-interface** property, referencing the head-end IP interface. The head-end presence contained within the SIP configuration results in the parent IP interface being treated by AA-SBC as a backing interface, and the reference as the head-end. An interface becomes a backing interface when it contains a pointer to a head-end interface. Do this for each interface that is to be a backing to a head-end interface.

To correctly configure load-balancing for SIP processing, you must do the following:

1. Configure the ip interfaces that will be used for both the headend and backing interfaces.
2. The sip properties of the backing interfaces must match those of the head interface. For example, they must all use the same port assignments, and if you are using TLS, they must all use the same certificate.

3. You must enable the master services registration object so that the interfaces can share the registration database.
4. You must enable the master services load-balancing object so that the interfaces can share the rules database.

To verify your configuration, first ensure that all sip properties match. From the CLI of the box that hosts the headend, execute the **show load-balance** command. This lists all associated backing interfaces (and statistics). From each box hosting a backing interface, execute **show backing-interface** to display configuration and statistics information.

Load-balancing with hash-based mode

With hash-based load balancing, AA-SBC does not create a rule for each connection. Instead, it uses a hash function to ensure that all traffic on a connection gets forwarded to the same backing interface. The hash is calculated on the source IP address and port of the incoming traffic. Since this information does not change during the lifetime of a connection, AA-SBC will always forward traffic to the same backing interface, without the need for a separate rule.

Configuring Messaging Client Tunnels

In some cases, you may want AA-SBC to act as a proxy between a messaging client and its native server. To do this, you set up a listening port (for the appropriate transport protocol) on AA-SBC, and when it receives traffic on that port from the configured messaging client, it forwards it to configured partner server. To set up this kind of tunnel, do the following:

1. Configure the client as a **sip-host** server. Set the server-type to the messaging client type (typically **office-communicator** or **nortel-mcp**). The client is the “from server.” Leave the **domain** property of the **server** configuration blank.
2. Configure the native **server** (for example, **lcs** or **mcs**). This is the “to server.”
3. Set the listening port on AA-SBC using the **udp-port**, **tcp-port**, or **tls-port** properties of this object. Assign your configured to and from servers.
4. If you want matching policy to modify the SIP message when messaging client tunnels are configured, enable the **tunnel-policy** property in the vsp settings object. (By default, messages passing through these client tunnels are not subject to policy modifications.)

Syntax

```
config cluster box number interface ethX ip name sip load-balancing
config cluster vrrp vinterface vxID ip name sip load-balancing
```

Properties

Property name	Description
hash-function {source-address-and-port | source-address-low-octet | source-address | source-address-port-and-protocol | source-port}	Sets the hash method to use to ensure that all traffic on a connection gets forwarded to the same backing interface. The values used in the hash function are derived from the IP header on the Ethernet/IP frame. Select a method for the system to base the calculation on: • **source-address-and-port**—the source IP address and port value. • **source-address-low-octet**—the bottom seven bits of the source port value. • **source-address**—the source IP address value. • **source-address-port-and-protocol**—the source IP address, port, and protocol values. • **source-port**—the bottom seven bits of the source port value. **Example: set hash-function rule-based** The default setting is **source-address-and-port**.
head-end-interface *interfaceReference*	Specifies a head-end interface to serve as the central distribution point of SIP traffic. The parent object of this setting becomes, by definition of the configuration, a backing interface. A backing interface can support only one head-end interface. See Configuring head-end and backing interfaces for rules on configuring the head-end interface correctly. **Example: set head-end-interface “cluster vrrp vinterface vx1 ip headend1”** There is no default setting.

66. SNMP objects

SNMP description

AA-SBC supports remote management access using the Simple Network Management Protocol (SNMP). SNMP is the Internet standard remote management protocol for network devices. Running a remote SNMP application (the SNMP manager) from a PC or workstation, you can communicate with the SNMP component on AA-SBC (the SNMP agent) to retrieve information about manageable objects on the appliance as well as edit the appliance configuration settings.

AA-SBC imports into the enterprise MIB (Enterprise MIB objects—cxc.mib):

- SNMPv1-SMI
- SNMPv2-TC
- Standard MIB-II objects (RFC 1213-MIB)

And supports:

- GET, GETNEXT, and SET requests
- TRAP commands.

For a more detailed description of SNMP, see the *Net-Net OS-E – System Administration Guide*.

SNMP object summary

The following table lists and briefly describes the **snmp** objects. See the following chapters for other objects in the CLI hierarchy:

- Chapter 14, “Cluster, box, and interface objects”
- Chapter 77, “VLAN objects”
- Chapter 35, “IP objects”

Object name	Description
snmp	Configures the AA-SBC SNMP property settings.

snmp

Purpose

Configures the SNMP protocol on AA-SBC.

Syntax

```
config cluster box number interface ethX ip name snmp
config cluster box number interface ethX vlan number ip name snmp
config box interface ethX ip name snmp
config box interface ethX vlan number ip name snmp
```

Properties

Property name	Description
admin {enabled | disabled}	Sets the administrative state of the SNMP protocol, either **enabled** (running) or **disabled**. When disabled, you can still configure the SNMP parameters, but the settings do not become active until the **admin** property is set to **enabled**. **Example: set admin enabled** The default setting is **enabled**.
port *integer*	Sets the UDP port over which the system listens for SNMP messages. UDP port 161 is the known port for SNMP messages; UDP port 162 is the known port for SNMP trap messages. **Example: set port 161** Enter a port number between 1 and 65535. The default SNMP UDP port is **161**.
version {1 | 2c | 3}	Sets the version of SNMP to run between the remote SNMP manager and the system SNMP agent; either SNMP V1, V2C, or V3. **Example: set version 2c** The default SNMP version is **2c**.

Property name	Description
community *string*	Sets the SNMP community string that allows the remote SNMP management system to access AA-SBC management information and statistics. The SNMP community string is similar to a user id or password that is included with all SNMP requests. If the community string is correct, the CMS Desktop responds to the SNMP request. If the community string is incorrect, the system discards the SNMP request. Enter the SNMP community string using up to 32 alphanumeric characters with no blank spaces. **Example: set community private** The default community string is **public**.
trap-target *ipaddress* [*port*]	Sets the IP address and optional TCP port of one or more remote hosts to which the system sends SNMP traps. You can configure up to 10 remote trap destinations. **Example: set trap-target 210.123.10.8** There is no default IP address setting. The default TCP port is 162 if not specified. TCP port 162 is the known port for SNMP trap messages.
trap-retransmit {enabled *minutes* | disabled}	Specifies whether the system should continue to retransmit SNMP traps until it receives an acknowledgement from the trap target. Use the trap-reset action to send the acknowledgement. If **enabled**, set the interval, in minutes, between retransmissions. **Example: set trap-retransmit enabled 30** The default setting is **disabled**.
trap-filter {generic | csta | dos | sip | system | tls}	Specifies which categories of SNMP traps the system sends to the remote host(s) configured with the **trap-target** property. You can set as many of the pre-configured trap categories as necessary. If you do not set any trap filters, the system sends all traps. Use the **show trap-categories** command to list the possible trap types in each category. **Example: set trap-filter sip** There is no default setting.

67. Static stack settings objects

Static stack settings description

The properties within the static-stack-settings object are all configuration settings that cannot be changed dynamically. Any changes to these properties do not take effect until you issue a vsp-reset action or restart AA-SBC.

Static stack settings object summary

The following table lists and briefly describes the **static-stack-settings** object. See the following chapter for other objects in the CLI hierarchy:

- Chapter 75, “Virtual system partition (VSP) objects”

Object name	Description
`static-stack-settings`	Modifies static values.

static-stack-settings

Purpose

Modifies static properties of the VSP. Any changes to these properties do not take effect until you issue a vsp-reset action or restart AA-SBC.

AA-SBC uses the standard T1, T2, and T4 SIP timers, as described in *RFC 3261, SIP: Session Initiation Protocol*. These timers are the source for all other CXC timer settings in the SIP stack. There may be cases when you want to change the setting of these source times, for example to shorten connection times when memory is running low or to configure AA-SBC to wait longer before initiating a retransmission.

Syntax

```
config vsp static-stack-settings
```

Properties

Property name	Description
domain-name *URL*	Sets the DNS domain name for the VSP. The registration service only accepts registration requests matching this domain name. **Example: set domain-name company.com** There is no default setting.
domain-alias *string*	Adds one or more aliases for the domain in which the system resides. This is useful in cases where a SIP server uses an address or other identifier instead of a domain name or where two interfaces on a CXC are configured to receive REGISTER requests. Adding an identifier—an IP address or a string—to the alias list allows the system to accept requests addressed to that identifier. **Example: set domain-alias 198.162.10.10** There is no default setting and no limit to the number of aliases that can be added.

Property name	Description
location-lookup-pattern {whole-uri | user-host | user-only}	Specifies what portion of the URI to use when doing location cache lookups. Because vendors can change the URI format, it is difficult to maintain a consistent location cache. The system creates a URI alias table, which indexes various URI formats to an AOR. Use this property to set the portion you want the system to consider, either: • **whole-uri**—the entire body of the URI. If the URI is a TEL URI, the system only considers the user portion. • **user-host**—the user and host portion of the URI. • **user-only**—the user portion of the URI. Note that you must execute the vsp-reset action if you change the lookup pattern. If you do not, the alias table will be corrupt. **Example: set lookup-pattern user-host** The default setting is **whole-uri**.
stack-worker-threads-max {automatic | *threads*}	*Secondary property*. Sets the number of SIP stack processing threads to create for this VSP. The system can support multiple execution threads, each concurrently working on a different SIP message. It is useful to configure extra threads to run, since some threads may occasionally block, for example while waiting for a response from an authentication server. If you set the maximum to 0, the system executes a single thread. **Example: set stack-worker-threads-max 60** The default value for this property is **automatic**. See Using automatic values for more information.

Property name	Description
max-number-of-sessions {automatic | *integer*}	*Secondary property*. Sets the maximum number of concurrent SIP sessions that the VSP can support. This value includes all REGISTER, SUBSCRIBE, INVITE, and other sessions.the system allocates resources at boot up based on this number. (The vsp **cac-max-number-of-calls** property creates a dynamic value for call admission control.) **Example: set max-number-of-sessions 1500** The default value for this property is **automatic**. See Using automatic values for more information.
t1 *milliseconds*	Sets the value of the SIP T1 timer. The T1 timer, according to RFC 3261 is: "For unreliable transports (such as UDP), the client transaction retransmits requests at an interval that starts at T1 seconds and doubles after every retransmission. T1 is an estimate of the round-trip time (RTT), and it defaults to 500 ms." The T1 full timer description can be found in section 17.1.1.1 of RFC 3261 *RFC 3261.* **Example: set t1 1500** Enter a value from 10 to 10,000 milliseconds; the default setting is **2000** ms.
t2 *milliseconds*	Sets the value of the SIP T2 timer. The T2 timer, according to RFC 3261 is: "... the amount of time a non-INVITE server transaction will take to respond to a request, if it does not respond immediately." The T2 full timer description can be found in section 17.1.2.2 of *RFC 3261.* **Example: set t2 3000** Enter a value from 1,000 to 10,000 milliseconds; the default setting is **4000** ms.
t4 *milliseconds*	Sets the value of the SIP T4 timer. The T4 timer, according to RFC 3261 is: "the amount of time the network will take to clear messages between client and server transactions." The T4 full timer description can be found in section 17.1.2.2 of *RFC 3261.* **Example: set t4 4500** Enter a value from 1,000 to 10,000 milliseconds; the default setting is **5000** ms.

Static stack settings objects

Property name	Description
max-udp-session-linger *milliseconds*	Sets the number of milliseconds that the system maintains the internal data structure for a SIP transaction after its apparent useful life is over. (Keeping a session live makes it available in the event that a retransmission arrives.) You may want to reduce the session linger time if, for example, you are low on memory. **Example: set max-udp-session-linger 20000** Enter a value from 0 to 60,000 milliseconds; the default setting is **30,000** ms (30 seconds). A value of 0 sets the system to remove the session immediately on receipt/transmission of first final response.
max-udp-tunnel-inactivity *seconds*	Specifies the number of seconds that a UDP-to-UDP tunnel can remain inactive before it becomes eligible to be torn down. Because UDP is a connectionless protocol, it does not provide the cleanup utilities for idle sessions that a protocol such as TCP provides. Instead, the system defines inactivity with this property, and configures tunnel reclamation in the settings object (**udp-tunnel-reclaim** and **udp-tunnel-reclaim-scan interval** properties). **Example: set max-udp-tunnel-inactivity 7200** Enter a value from 300 to 60,000 seconds; the default setting is **3600** seconds.
max-proxy-transactions-per-vsp	*Secondary property*. Sets the maximum number of concurrent proxy transactions for the entire VSP. You can specify an integer or use the default, **automatic**, to use the platform specific factory value. **Example: set max-proxy-transactions-per-vsp** The default setting is **automatic**.

Static stack settings objects

68. STUN server objects

STUN description

Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs) (STUN), described in RFC 3489, enables SIP clients to discover the presence and types of NATs and firewalls that exist between them and the public Internet. In addition to providing a way for an application to traverse a NAT, the protocol provides for a connectivity check between a client and server separated by a NAT. It helps prevent NAT-associated network application failures by transmitting exploratory STUN messages, over UDP port 3478, between the server and clients.

STUN allows a client to not only determine its publicly addressable IP interface and port, but to set criteria for keeping those NAT bindings open. For example, a VoIP phone or software package may include a STUN client, which will send a request to a STUN server. The server then reports back to the STUN client with:

- the public IP address of the NAT router.
- the port opened by the NAT (for that client) to allow traffic back in to the network.

AA-SBC as a STUN relay server

AA-SBC provides a STUN/TURN server integrated into the signaling and media proxy architecture. The TURN server provides support for straight UDP relays, as well as for TCP-to-UDP conversion. AA-SBC supports both standard STUN (with binding discovery) and the newer version (with connectivity check usage).

As a STUN server, AA-SBC receives STUN requests from clients and responds. (The STUN client uses DNS to find AA-SBC/STUN server.) AA-SBC identifies the public-side NAT details by inspecting exploratory messages from STUN-enabled clients, sent to determine which transmit and receive UDP port to use. The AA-SBC STUN server examines the incoming message and informs the client which public IP address and ports were used by the NAT. These are then used in the call establishment messages sent to the SIP destination server.

The client may also send a TURN allocate request. AA-SBC finds an unused port on the relay-interface and sends a message back to the client with the port and IP address. The client then sends data to AA-SBC, which forwards the data to its final destination, changing the packet to look like it originated from that port on AA-SBC, which forwards return packets back to the client.

For complete information on STUN and TURN refer to:

- *RFC 3489—STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)*
- *draft-ietf-behave-rfc3489bis-05—Simple Traversal Underneath Network Address Translators (NAT) (STUN)*
- *draft-ietf-behave-turn-01—Obtaining Relay Addresses from Simple Traversal of UDP Through NAT (STUN)*

Understanding STUN configuration

The following discussion only applies to RFC 3489 (standard STUN). The updated STUN protocol (RFC 3489bis, as no RFC has yet been assigned to it) does not apply to this discussion. The discussion only applies if clients in your network are using advanced features of RFC 3489 STUN.

As a STUN server, AA-SBC listens on configured interfaces for STUN client requests. When using STUN for *Binding Discover*y, the client sends a *Binding Request* packet to the STUN server, and the server responds with a Binding Response. This response indicates the IP address and port the packet seemed to originate from (which may be different from the address/port the client sent the packet from, if there is an intervening NAT device). This address is often known as the *public address*, or *NAT mapping*. The client can then use that public IP address when registering with AA-SBC.

Some NAT implementations base their mappings not only on the client IP address and port, but also on the server IP address and port. A packet sent from the same client address/port to a different server (or even a different port on the same server) may be given a different NAT mapping by the NAT device. Any address information learned by doing Binding Discovery with the STUN server is unusable by other devices. To determine whether or not this is the case, a STUN Binding Request can request that the STUN response be sent from a different address (presumably a different interface on the same machine), different port, or both. For the same reason, when the server sends the Binding Response, it adds information indicating which address/port it would use if the client had asked for a response from a different address/port. The client can then use this information to send a new Binding Request to the alternate address/port.

For STUN to operate properly, follow these rules when configuring STUN servers:

- Create STUN server instances in pairs.
- Put each instance of the pair on a different IP address.
- Assign exactly two UDP ports to each; the port number assignments must be identical for each.
- The secondary interface (configured in the stun-server object) of each instance must point at the IP address of the other instance.

For example, with a STUN server configured on interface A, ports 100 and 200, configure an additional STUN server on interface B, ports 100 and 200. In the interface A configuration, set the **secondary-interface** property to B, and vice versa.

STUN-server object summary

The following table lists and briefly describes the **stun-server** objects. See the following chapters for other objects in the CLI hierarchy:

- Chapter 14, “Cluster, box, and interface objects”
- Chapter 77, “VLAN objects”
- Chapter 35, “IP objects”

Object name	Description
`stun-server`	Configures STUN protocol communications settings.

stun-server

Purpose

Configures the STUN and TURN server functionality on AA-SBC.

Syntax

```
config cluster box number interface ethX ip name stun-server
config cluster box number interface ethX vlan number ip name
   stun-server
config box interface ethX ip name stun-server
config box interface ethX vlan number ip name stun-server
```

Properties

Property name	Description
admin {enabled | disabled}	Sets the administrative state of the STUN protocol, either **enabled** (running) or **disabled**. **Example: `set admin enabled`** The default setting is **enabled**.
port {UDP | TCP | TLS} *portNumber* [enabled | disabled]	Specifies the protocol and port number over which STUN messages are transmitted between a SIP client and the system STUN server. In addition, you can specify whether to enable the TURN-redirector. When **enabled** for a port, TURN Allocate requests to that port may be redirected to another port on another box in the cluster (using the 300 Redirect error response). If no other systems in the cluster are configured to share TURN ports, or if the local machine is the best choice, the request is fulfilled locally. When the TURN-redirector is **disabled**, requests to this port are fulfilled locally or fail. If using the TURN-redirector, you must set the cluster **share-turn-ports** property to true to determine which systems have their TURN ports available as a target for redirection. **Example: `set port UDP 3478`** The default protocol for STUN is **UDP**; the default port is **3478**. Enter a value between 1 and 65535. Turn-redirector is **disabled** by default.
certificate *certificateReference*	Specifies the certificate used to connect to the STUN server over TLS. This is the certificate that the system presents to the client. **Example: `set certificate vsp tls certificate my NetworkCert.pfx`** There is no default setting.

Property name	Description
stun-auth-level {ignore | allow | short-term}	Specifies the level of authentication to require from the client for STUN requests when the transport protocol used is not TLS. (Note that this setting does not apply to TURN requests, which always use long-term authentication.) Select one of the following: • **ignore**—the system does not require authentication from the client, and ignores any authentication presented. Even if the authentication is incorrect, when set to ignore, the system accepts the request. • **allow**—the system does not require authentication from the client, but if it is presented, verifies it. If the authentication is correct or there is none, the system accepts the request. If it is incorrect, the system refuses the request. • **short-term**—the system requires the client to use authentication that has short-term credentials. The client requests a shared secret from the system, via a TLS connection, which is given in the form of an automatically generated username and password. These credentials are used for a fixed time period of either nine minutes (when used outside the context of a TURN allocation), or for the lifetime of the allocation (when used inside the context of a TURN allocation). **Example: `set stun-auth-level short-term`** The default setting is **allow**.
short-term-user-secret *passwordTag*	Sets a private secret used to seed the random number generator for generated short-term credentials. This value does not need to be known by the client. If you do not configure this value, the system uses a fixed default value. If multiple systems use TURN servers cooperating in a cluster, they must all be configured with the same **short-term-user-secret** setting. **Example: `set` short-term-user-secret pswd1** There is no default setting.

Property name	Description
secondary-interface *interfaceReference*	Specifies the interface to use when the CHANGE-REQUEST attribute requests a response from a different IP address. The interface you specify as the secondary interface must: • have a STUN server configured, • use the same port number assignments as the primary interface • have a secondary-interface that points back to this interface. See Understanding STUN configuration for “old-style STUN” configuration requirements. **Example: `set` secondary-interface `“cluster box 1 interface eth0 ip a”`** There is no default setting.
allow-turn {enabled | disabled}	Enables or disables TURN on the system STUN server. **Example: `set` allow-turn disabled** The default setting is **enabled**.
relay-interface *interfaceReference*	Specifies the interface over which the client receives public visibility; the interface from which the system allocates TURN relay ports. Create a reference to a configured interface. The interface that you select must have media-ports enabled and a port pool range defined, but does not require a STUN server instance configured. **Example: `set` relay-interface `“cluster box 1 interface eth1 ip z”`** There is no default setting.
allocation-lifetime-max *seconds*	Specifies the maximum number of seconds that a TURN relay port allocation remains valid. Prior to expiration, the client must send a reallocation (renewal) request. The client can also send a request to immediately close the port. **Example: `set` allocation-lifetime`-max 4200`** Enter a value between 1 and 100,000; the default setting is **3600** seconds.

Property name	Description
allocation-bandwidth-max *kbps*	Specifies the maximum amount of bandwidth that can be granted in a TURN relay port allocation. If a request to the STUN server exceeds the allocation, the system returns a response indicating this value. **Example: `set allocation-bandwidth-max 750`** Enter a value between 1 and 1,000,000; the default setting is **500** kbps.
allocation-bandwidth-default *kbps*	Specifies the amount of bandwidth allotted to a TURN relay port if the client did not request a specific amount. **Example: `set allocation-bandwidth-default 300`** Enter a value between 1 and 1,000,000; the default setting is **150** kbps.
ta *ms*	Sets the duration of the Active Destination state transition timer (as defined in draft-ietf-behave-turn-01.txt, section 8.3). This timer is used in cases when the client sets an active destination, so that all subsequent data received from the active peer is forwarded without STUN encapsulation to the client (and vice versa) and then later sets a different active destination. This property sets the amount of time, during the switch from old to new active destination, that forwarded traffic is encapsulated inside STUN Data Indications (to avoid confusion about the destination from which it came). Leave this value set to the default unless you have pressing reason to change it. **Example: `set` ta 3000** Enter a value between 0 and 10,000; the default setting is **3000** milliseconds.

69. Telnet objects

Telnet description

Telnet is the standard TCP/IP-based terminal emulation protocol defined in *RFC 854, Telnet Protocol Specification*, that provides a standard method for local and remote terminal communications over an IP network.

AA-SBC uses Telnet to establish a connection to the CLI. The Telnet objects allow you to configure the parameters of the Telnet session.

Basic Telnet configuration

The following table describes the steps for connecting to the CLI over Telnet. For more detailed information about opening a Telnet session and starting the CLI, see Chapter 1, "Using the Net-Net 2600 Command Line Interface."

Step	Action
1.	Ensure that AA-SBC is connected to a network that the remote system can reach.
2.	Specify the Ethernet interface to be used for Telnet sessions and create a named IP configuration on that interface.
3.	Start the Telnet client at the local or remote terminal.
4.	Log in and start the CLI.

Telnet object summary

The following table lists and briefly describes the **telnet** object. See the following chapters for other objects in the CLI hierarchy:

- Chapter 14, “Cluster, box, and interface objects”
- Chapter 77, “VLAN objects”
- Chapter 35, “IP objects”

Object name	Description
`telnet`	Configures the system Telnet communications settings.

telnet

Purpose

Configures the Telnet protocol on AA-SBC.

Syntax

```
config cluster box number interface ethX ip name telnet
config cluster box number interface ethX vlan number ip name telnet
config box interface ethX ip name telnet
config box interface ethX vlan number ip name telnet
```

Properties

Property name	Description
admin {enabled | disabled}	Sets the administrative state of the Telnet protocol, either **enabled** (running) or **disabled**. When disabled, the parameters of Telnet can still be configured, but do not become active until **admin** is set to **enabled**. **Example: set admin disabled** The default setting is **enabled**.
max-sessions *integer*	Sets the maximum number of simultaneous Telnet sessions allowed. **Example: set max-sessions 4** Enter a value between 1 and 32. The default number of sessions is **8**.
idle-timeout *integer*	Specifies the amount of time (in seconds) allowed to elapse before the system closes the Telnet session due to inactivity. **Example: set idle-timeout 120** Enter a value between 60 and 86,400. The default setting is **600** seconds (10 minutes).
port *integer*	Identifies the Ethernet port through which the system listens for Telnet sessions. **Example: set port 21** Enter a port number between 1 and 65535. The default Telnet port is **23**.

70. TFTP server objects

TFTP description

AA-SBC uses a Trivial File Transfer Protocol (TFTP) server to upload and download executable images and configurations between AA-SBC and other devices, such as IP phones. You configure TFTP servers on Ethernet and VLAN interfaces.

TFTP server object summary

The following table lists and briefly describes the **tftp** object. See the following chapters for other objects in the CLI hierarchy:

- Chapter 14, “Cluster, box, and interface objects”
- Chapter 77, “VLAN objects”
- Chapter 35, “IP objects”

Object name	Description
`tftp`	Opens the TFTP server configuration object for editing.

tftp

Purpose

Opens the TFTP server configuration object on AA-SBC and sets the administrative state and the port to which clients send and receive files. In addition, you can specify the directory to which AA-SBC writes TFTP transfers.

Syntax

```
config cluster box number interface ethX ip name tftp
config cluster box number interface ethX vlan number ip name tftp
config box interface ethX ip name tftp
config box interface ethX vlan number ip name tftp
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables the TFTP server on the system. **Example: set admin enabled** The default setting is **enabled**.
port *integer*	Specifies the system port on which the TFTP server is listening for file transfer requests. **Example: set port 30** The default TFTP port is **69**. Enter a value between 1 and 65535.
directory *pathName*	Specifies the default directory to which the system writes TFTP transfers. **Example: set directory /cxc_common/tftp_transfers/east** The default directory path is **/cxc_common/tftp**.

71. Third-party call control server objects

3PCC server description

The AA-SBC can act as an intermediary between third-party call control (3PCC) servers (e.g., BroadWorks, Cisco Call Manager and Avaya AES) and Microsoft OCS. The AA-SBC communicates with OCS using CSTA-over-SIP. (CSTA is a protocol used by OCS to communicate call state information, which can then be reflected in user presence status.) The AA-SBC communicates with the 3PCC servers using OCI (BroadWorks) or JTAPI (Cisco and Avaya). You must configure the jtapi master-service to enabled third-party call control.

Third Party Call Control functionality enables Windows Communicator clients to place calls between registered SIP telephony endpoints using a 3PCC server and to place calls between any PSTN-enabled phones (wired line, mobile phones, etc.) via standard SIP gateways. In addition, subscribers can take advantage of features such as the ability to: place calls on hold; conference in third parties; transfer calls; use click-to-call dialing when calling other parties; single click to begin a conference call; accept incoming calls to the desk phone using the Communicator client; and redirect calls to any other desktop or mobile telephone.

Computer-supported telecommunications applications (CSTA) is a third-party call control protocol. It provides an abstraction layer for telecommunications applications, independent of underlying signaling protocols and devices. CSTA supplies applications services that allow a user agent to observe and control media calls (voice, IM, email, etc.). Microsoft implements SIP-based CSTA in its OCS application.

CAP is a proprietary call control interface used by Broadsoft in their SIP Application Servers to control BroadWorks calls remotely. It is an HTTPS-based protocol. When CSTA mode is set to OCI, the AA-SBC acts as a translator, allowing an OCS to view the Broadsoft BroadWorks PBX server as CSTA user agent. The LCS can then use “first-party” call control from MS Office Communicator to SIP phones in the enterprise. Communicator can originate, answer, and transfer calls (as well as execute other features) to the Broadsoft servers with the AA-SBC in between providing the conversion.

The AA-SBC supports several 3PCC server/PBX options. These include the BroadWorks, Cisco, and Avaya IP telephony platforms. The AA-SBC supports configuration to communicate with the ITC TCS server. Using this server, users can take advantage of the voice drop capabilities available through desktop environment APIs and the call-control action. In addition, you can configure an internal 3PCC server. An internal 3PCC server configures the AA-SBC to act as the PBX, resulting in phones registering with the AA-SBC. This mode only works with phones registered directly to with the AA-SBC. Also, you can set up a loopback configuration for testing purposes. the AA-SBC tests connectivity between itself and the OCS server.

CLI hierarchy information

See the following chapters for other objects in the CLI hierarchy:

- Chapter 75, “Virtual system partition (VSP) objects”
- Chapter 27, “Enterprise objects”

3PCC server object summary

The following table lists and briefly describes the **3pcc-servers** objects.

Object name	Description
`3pcc-servers`	Configures the parameters for communication between the third-party call control server and the AA-SBC.
`cisco-call-manager-server`	Specifies JTAPI login parameters for the Cisco Call Manager.
`aes`	Specifies JTAPI login parameters for the Avaya AES.
`backup-host-name`	Adds a backup PBX for either the Cisco CallManager or Avaya AES.

3pcc-servers

Purpose

Opens the server configuration object to allow setting the parameters for communication between the AA-SBC and the third-party call control server. The AA-SBC supports the following 3PCC servers:

- Internal CSTA server
- Broadworks CSTA server, either OCI or OCS
- Cisco Call Manager
- Avaya Communications Manager
- Loopback CSTA server for testing
- IPC CTS server

Note: While you can configure third-party call control servers at any time, you must enable the master-services jtapi object for the AA-SBC device to use the server. See Chapter 39, “Master services objects” for more information.

Setting IPC server line IDs

When using the IPC CTS server with desktop APIs and the AA-SBC, you must configure a prefix and pool of available numbers used to create a unique line ID for each call. The numbers you set within this object should also be configured on the IPC server and used by it for outgoing calls to the AA-SBC. The configuration through this object allows the AA-SBC to recognize certain numbers as belonging to the IPC PBX, and can then do the appropriate mapping.

To configure the number set, first set the **number-prefix** property. Enter a string to identify the prefix, for example: 1-978-555 or tel:+555. Next you define the pool of numbers available by setting the **range-min-number** and **range-max-number**. From this pool, the IPC server will assign an extension to the prefix that is valid for the duration of the call. When the call terminates, that number is once again available. The pool can be any range or length, as long as it is configured similarly for both ends.

Syntax

```
config vsp enterprise 3pcc-servers internal-csta-server string
config vsp enterprise 3pcc-servers broadworks-csta-server string
config vsp enterprise 3pcc-servers cisco-call-manager string
config vsp enterprise 3pcc-servers avaya-communications-manager string
config vsp enterprise 3pcc-servers loopback-csta-server string
config vsp enterprise 3pcc-servers ipc-server string
```

Properties

Property name	Description
`description` *`string`*	Associates a text string with a server configuration. The string displays in some event logs and status providers to help identify the target. **Example: set description callMgrServer** There is no default setting.
`admin {enabled | disabled}`	Specifies whether the system uses this server in the current session. If **enabled**, the system uses this server. If **disabled**, the system does not use this server. **Example: set admin enabled** The default setting is **enabled**.

`inbound-uri-normalization`
`regEx replacement`
	This property is only applicable to Cisco and Avaya call managers. Translates received Cisco or Avaya 3PCC server phone numbers into a OCS/MOC-acceptable form. The AA-SBC normalizes requests from the 3PCC server that are destined for a Microsoft OCS server using the AA-SBC. When the AA-SBC communicates with OCS/MOC, the numbers must be in an OCS-acceptable form (e.g., tel:+12126474840). Because the 3PCC server responds with and understands numbers in a different form (e.g., 6474840), the AA-SBC must perform a translation (prepending “tel:+1212”) before forwarding the request.
	Example: set inbound-uri-normalization ^([0-9]{7})$ tel:+1978\1
	There is no default setting.
	For more information regarding configuring regular expressions and replacement strings, see the section in Chapter 1, “Using regular expressions” on page 36.

Property name	Description
`outbound-uri-normalization` *`regEx replacement`*	*This property is only applicable to Cisco, BroadWorks, and Avaya call managers*. Translates phone numbers received from the Microsoft OCS server into an acceptable form for the Cisco, BroadWorks, or Avaya 3PCC server. The AA-SBC normalizes requests from the OCS server that are destined for a 3PCC server via the AA-SBC. When the AA-SBC communicates with OCS/MOC, the numbers must be in an OCS-acceptable form (e.g., tel:+12126474840). Because the 3PCC server responds with and understands numbers in a different form (e.g., 6474840), the AA-SBC must perform a translation (stripping “tel:+1212”) before forwarding the request. **Example: set outbound-uri-normalization ^tel:(\+)?((1?508)[0-9]{7}) 9\2** There is no default setting. For more information regarding configuring regular expressions and replacement strings, see the section in Chapter 1, “Using regular expressions” on page 36.

Property name	Description
`server-uri-normalization` *`regEx replacement`*	*This property is only applicable to Cisco and Avaya call managers*. Translates phone numbers coming into the AA-SBC from a 3PCC server. This normalization is used when the call manager changes a number before responding to the AA-SBC. The AA-SBC does a “first-pass” normalization of the number before applying the inbound-uri-normalization replacement and sending the call to OCS/MOC. For example, the AA-SBC may send 915085551212 when placing a call, to which the call manager may respond with 15085551212. This normalization would replace the “9” before forwarding the call. **Example: set server-uri-normalization ^9([0-9]{11})$ \1** There is no default setting. For more information regarding configuring regular expressions and replacement strings, see the section in Chapter 1, “Using regular expressions” on page 36.
`moc-keepalive-timeout` *`minutes`*	Specifies the number of minutes the system waits for a keepalive message from a MOC client. The client can be set to send a keepalive at a configured interval (configured at the client side). This value must be set to a value higher than the client setting. If the system does not receive a keepalive from the client within the time set with this property, it sends a message to the event log indicating a keepalive timeout. A setting of 0 (the default) disables this feature. **Example: set moc-keepalive-timeout 10** Enter a value between 0 and 60: the default setting is **0** minutes.
broadworks-csta-server only `server` *`host`*	Specifies the BroadWorks application server to which the AA-SBC connects in order to provide 3PCC services. Enter a hostname or IP address to identify the server (PBX). **Example: set server 172.46.20.231** There is no default setting.

Property name	Description
broadwoks-csta-server only `type {oci | ocs} port`	Specifies the type of BroadWorks PBX being used to connect calls. Also, set the port number over which the BroadWorks server is listening (the port to which the AA-SBC connects). Select either: • **oci**—the system acts as a translation device, converting CSTA traffic to a format the OCI server (e.g., BroadWorks PBX) can recognize. The AA-SBC sends BroadWorks CAP authentication messages to the Broadworks OCI server. The BroadWorks PBX connects the calls. • **ocs**—the system acts as a translation device, converting CSTA traffic to a format the OCS server (e.g., BroadWorks PBX) can recognize. The AA-SBC sends BroadWorks OCI and CAP authentication messages to the Broadworks OCS server. The BroadWorks PBX connects the calls. **Example: set type oci** The default type setting is **oci** with a listening port of **2206**.

Property name	Description
broadwarks-csta-server only `oci-user` *`regEx replacement`*	Specifies the name that will be used to log in to the BroadWorks server. This property creates the conversion strategy to map a domain name into a BroadWorks-acceptable format. • *regEx*—enter a regular expression identifying the portion of the attribute to match. For example, the following expression identifies a subexpression (between the parenthesis) that matches all names: (.*) • *replacement*—enter a string that defines how to recompose the resulting regEx string. In the following example, the first component from the regular expression is substituted in place of the “1” and appended to “@company.com.” \1@company.com **Example: set oci-user (.*)@company.com(.*) “\1@test.com\2”** There is no default setting. For more information regarding configuring regular expressions and replacement strings, see the section in Chapter 1, “Using regular expressions” on page 36.
cisco-call-manager only `version {cisco4.0 | cisco5.0 | cisco6.0}`	Specifies the version of the Cisco Call Manager that is running on the server to which the AA-SBC is connecting. **Example: set version cisco5.0** The default setting is **cisco4.0**.

Property name	Description
cisco-call-manager and **avaya-communications-manager** only `check-jtapi-connection {true | false}`	Specifies whether to verify the JTAPI connection. If set to true, when the system receives a MOC keepalive from the client, it verifies that the connection to the JTAPI server is still available. If the connection is not up, the system logs an event message. **Example: set check-jtapi-connection true** The default setting is **false**.
avaya-communications-manager only `jtapi-debug {true | false}`	Controls debug logging for the JTAPI library. When set to **true**, the JTAPI library creates a log file on the system and writes log messages to that file. **Example: set version cisco5.0** The default setting is **false**.
avaya-communications-manager only `jtapi-debug-level {cisco4.0 | cisco5.0 | cisco6.0}`	Specifies the message level to write to the JTAPI library log on the AA-SBC. This property is only applicable if the **jtapi-debug** property is set to **true**. Set a level, as defined below, and the system writes statements of that level to /cxc/avayaJtapiLog.OUT. Log levels are, from the JTAPI specification: • **0 NONE**—no debug statements • **1 SPY**—cstaSpy-like statements • **2 OBJECT**—SPY + create/delete/move TS objects • **3 HANDLER**—OBJECT + csta event handler info • **4 EVENT**—HANDLER + jtapi event info • **5 EXCEPTION**—EVENT + exception info • **6 INFO**—EXCEPTION + other interesting info • **7 ALL**—INFO + audit dump **Example: set jtapi-debug-level 7** Enter a debug level between 0 and 7; the default setting is **3**.

Property name	Description
ipc-cts-server only `host` *`host`*	Specifies the ITC application server to which the AA-SBC connects in order to provide 3PCC services. Enter a hostname or IP address to identify the server (PBX). **Example: set server 196.34.0.100** There is no default setting.
ipc-cts-server only `port` *`portNumber`*	Specifies the port on IPC server to which the AA-SBC connects. **Example: set port 5700** The default setting is port **5712**.
ipc-cts-server only `number-prefix` *`string`*	Specifies the prefix that identifies calls to be terminated at the system. The prefix is completed using the **number-postfix-min** and **number-postfix-max** properties, which with the prefix, define the entire range of numbers that can be used. The prefix can take whatever form is required for the desktop application. The AA-SBC matches on the string to determine that the call is intended to be terminated. **Example: set number-prefix 1-978-555** The default setting is port **5712**.
ipc-cts-server only `number-postfix-min` *`integer`*	Specifies the starting range for numbers appended to the prefix. See Setting IPC server line IDs for more information. **Example: set number-postfix-min 1** The default setting is port **5712**.

Property name	Description
ipc-cts-server only `number-postfix-max` *`integer`*	Specifies the end range for numbers appended to the prefix. See Setting IPC server line IDs for more information. **Example: set number-postfix-max 9999** The default setting is port **5712**.
loopback-csta-server only `call-delay` *`seconds`*	Specifies the number of seconds to wait, in the event of a non-response from the call manager being loopback tested, between the origination and termination of a call. **Example: set call-delay 15** The default setting is **10** seconds.

cisco-call-manager-server

Purpose

Configures the JTAPI login parameters between the AA-SBC and the Cisco Call Manager server. These authentication and configuration parameters set access and connection information between the AA-SBC and the switch hosting the Cisco Call Manager.

Only applicable to

- Cisco Call Manager

Syntax

```
config vsp enterprise 3pcc-servers cisco-call-manager name
   cisco-call-manager-server host
```

Properties

Property name	Description
`login` *`string`*	Specifies the login name that the Cisco Call manager server is configured to accept. **Example: set login admin** There is no default setting.
`password` *`string`*	Specifies the password that the Cisco Call manager server is configured to accept. **Example: set password admin** There is no default setting.
`connections` *`string`*	Specifies the number of sockets left open between the AA-SBC and the Cisco Call Manager server. **Example: set connections 4** Enter a value between 1 and 16; the default setting is **1**.
`provider-timeout` *`minutes`*	Specifies the number of minutes to wait before the system determines that the server is down or otherwise unavailable. If there is no connection or the connection fails, the system waits the amount of time specified in the **recovery-wait-time** property before testing the connection again. **Example: set provider-timeout 2** Enter a value between 1 and 60; the default setting is **1** minute.
`address-timeout` *`seconds`*	Specifies the number of seconds the system allows the PBX to return a verification of the address that logged in. This is the time to wait during a MOC login for the phone assigned to this user to come into service. **Example: set address-timeout 10** Enter a value between 1 and 60; the default setting is **4** seconds.

Property name	Description
`recovery-wait-time` *`minutes`*	Specifies the number of minutes between retries of the server connection. The system will continue retrying the connection until it is re-established or the time set with the **recovery-time-total** expires. The system begins retrying the server connection after the time set with the **provider-timeout** property has expired. **Example: set recovery-wait-time 2** Enter a value between 1 and 60; the default setting is **4** minutes.
`recovery-time-total` *`minutes`*	Specifies the total number of minutes that the system will spend retrying the connection to the server. The system waits the number of minutes specified in **recovery-wait-time** between retries. For example, if the wait time is 5 minutes, and you want the system to retry 5 times, set this property to 25. **Example: set recovery-time-total 120** Enter a value between 5 and 1440; the default setting is **60**.
`forwarding-url`	*Secondary property.* This property is a customer-specific application fix for a Cisco issue.
`forwarding-cluster`	*Secondary property.* This property is a customer-specific application fix for a Cisco issue.
`forwarding-login`	*Secondary property.* This property is a customer-specific application fix for a Cisco issue.
`forwarding-password`	*Secondary property.* This property is a customer-specific application fix for a Cisco issue.

aes

Purpose

Configures the JTAPI login parameters between the AA-SBC and the Avaya Application Enablement Services (AES) server. These authentication and configuration parameters set access and connection information between the AA-SBC and the switch hosting the Avaya Call Manager. The Avaya implementation requires a properties file that lists addresses of the AES servers. the AA-SBC creates and maintains this file. The JTAPI implementation requires a server name to use when contacting an Avaya AES. The properties of this object configure both.

Only applicable to

- Avaya Communications Manager

Syntax

```
config vsp enterprise 3pcc-servers avaya-communications-manager name
   aes host
```

Properties

Property name	Description
login *string*	Specifies the login name that the AES server is configured to accept. **Example: set login admin** There is no default setting.
password *string*	Specifies the password that the AES server is configured to accept. **Example: set password admin** There is no default setting.
connections *string*	Specifies the number of sockets left open between the AA-SBC and the AES server. **Example: set connections 4** Enter a value between 1 and 16; the default setting is **1**.

Property name	Description
`provider-timeout` *`minutes`*	Specifies the number of minutes to wait before the system determines that the server is down or otherwise unavailable. If there is no connection or the connection fails, the system waits the amount of time specified in the **recovery-wait-time** property before testing the connection again. **Example: set provider-timeout 2** Enter a value between 1 and 60; the default setting is **1** minute.
`address-timeout` *`seconds`*	Specifies the number of seconds the system allows the PBX to return a verification of the address that logged in. This is the time to wait during a MOC login for the phone assigned to this user to come into service. **Example: set address-timeout 10** Enter a value between 1 and 60; the default setting is **4** seconds.
`recovery-wait-time` *`minutes`*	Specifies the number of minutes between retries of the server connection. The system will continue retrying the connection until it is re-established or the time set with the **recovery-time-total** expires. The system begins retrying the server connection after the time set with the **provider-timeout** property has expired. **Example: set recovery-wait-time 2** Enter a value between 1 and 60; the default setting is **4** minutes.

Property name	Description
`recovery-time-total` *`minutes`*	Specifies the total number of minutes that the system will spend retrying the connection to the server. The system waits the number of minutes specified in **recovery-wait-time** between retries. For example, if the wait time is 5 minutes, and you want the system to retry 5 times, set this property to 25. **Example: set recovery-time-total 120** Enter a value between 5 and 1440; the default setting is **60**.
`name` *`string`*	Specifies the host name of the Avaya AES server. JTAPI requires this name when responding to CSTA requests. **Example: set name AV4** There is no default setting.

backup-host-name

Purpose

Configures a backup call manager for either the Cisco or Avaya platform. When the AA-SBC initiates contact, it indicates the primary PBX and if this is configured, a secondary PBX. The host names supplied here (and for the primary) must match those configured on the Cisco or Avaya device. The settings that are configured for the primary using the cisco-call-manager-server or aes objects are then applied to the backup PBX configured here.

Syntax

```
config vsp enterprise 3pcc-servers avaya-communications-manager name
   aes host backup-host-name host
config vsp enterprise 3pcc-servers cisco-call-manager name
   cisco-call-manager-server host backup-host-name host
```

Properties

None

72. TLS objects

TLS description

For networks running the Transport Layer Security protocol (TLS), you need to configure the certification file and the private key information required to pass SIP traffic.

TLS (sometimes referred to as Secure Socket Layer or SSL) is an encapsulation and cryptographic protocol that provides privacy and security between communicating applications over the Internet. AA-SBC uses TLS to authenticate SIP users and to encrypt/decrypt SIP traffic across participating real time collaboration networks and enterprise SIP applications.

Certificate presentation

TLS handles presentation of certificates differently for clients (initiators) and servers (responders). Usually in a TLS connection, only the server presents a certificate—the client is only allowed to present a certificate if it is requested to do so by the server. Typically, AA-SBC functions as a TLS server, and as such, presents a certificate to the peer. Occasionally, however, AA-SBC functions as the client, for example, when it connects to a Microsoft LCS server. In this case, the server presents the certificate to AA-SBC.

This operation will effect the **peer-certificate-verification** property setting of the certificate object. **If-presented** sets AA-SBC to request a certificate from its peer, but allows the connection if the peer does not present. **Required** ensures that the peer presents a certificate for connection. If the certificate entry is used when AA-SBC is a client, then **If-presented** and **Required** are equivalent. (This is because the server always presents a certificate.)

These two settings function differently when AA-SBC is answering the connection (is the server). In the **if-presented** case, if the peer (client) doesn't present a certificate, AA-SBC still allows the connection. If set to **required**, if the client doesn't present, AA-SBC terminates the call. Realistically, therefore, the **if-presented** setting makes sense only for a AA-SBC-as-server connection.

Certificate verification

AA-SBC has the ability to verify a peer's certificate. By default, this behavior is disabled; all peer certificates are accepted. When enabled, however, AA-SBC verifies a peer's certificate and rejects the connection if the certificate doesn't meet configured requirements. To verify, AA-SBC checks the following:

- The validity of the certificate's chain. It must be signed by a trusted Certificate Authority (CA), and must not have expired.
- Clearing the Certificate Revocation List (CRL). This list tracks those certificates that a CA has revoked. If any of the certificates in the chain presented to AA-SBC appear in the CRL, AA-SBC rejects the connection.
- The name of the host that is presenting the certificate. If the name does not match the name AA-SBC expects (as set in the **required-peer-name** property), then AA-SBC rejects the certificate, even if the chain is valid.

Certificate files and CA files can be in either PEM or PKCS#12 format. CRL files must be in PEM format. For a complete description of the TLS protocol, refer to the following RFCs:

- *RFC 2246, The TLS Protocol Version 1.0*
- *RFC 3261, Session Initiation Protocol (see Section 26.3.1)*

Using certificate vs. default-outgoing-settings

AA-SBC uses a certificate configuration to identify the certificate file and the characteristics of the certificate. There are two types of certificate configuration—a named certificate entry that can be applied to specific TLS connections and default certificate settings for use when a specific entry was not identified.

The entry created by the certificate object is used when AA-SBC functions as a server in a TLS connection. Or, it can be used in a AA-SBC-as-client setup, if you have configured the connection to use a specific certificate. For example, when you set the connection type to the LDAP server to TLS in the directory object, you are required to enter a named certificate.

The entry created by the default-outgoing-settings object is used when AA-SBC is a client with an unspecified certificate. For example, if you were to set the protocol that the DNS resolver server uses to TLS, you are not prompted for a certificate name. AA-SBC uses either:

- the certificate identified in the sip-settings object, if the session matched a configured policy.
- the **default-outgoing-settings** if the session did not match a configured policy or the policy did not have a certificate specified.

The certificate and **default-outgoing-settings** objects are otherwise the same; all certificate properties are described in the **default-outgoing-settings** object.

TLS object summary

The following table lists and briefly describes the TLS objects. See the following chapter for other objects in the CLI hierarchy:

- Chapter 75, “Virtual system partition (VSP) objects”

Object name	Description
`tls`	Opens the TLS configuration object for editing, where you can configure certificate and private key information.
`default-outgoing-settings`	Sets the certificate file, requirements, and passphrase to be used when a certificate is not specified.
`default-ca`	Creates a store of CA files to check against certificates.
`default-crl`	Creates a store of certificate revocation files to check against certificates.
`certificate`	Creates a certificate object that can then be used in other areas of the configuration.

tls

Purpose

Opens the TLS configuration object for editing. Through this object you can configure certificate and private key information.

Syntax

```
config vsp tls
```

Properties

None

default-outgoing-settings

Purpose

Creates a certificate object that sets characteristics of certificate use. This default certificate is used in cases when TLS is selected as a transport protocol and a specific certificate is not specified. See Using certificate vs. default-outgoing-settings for more information.

The certificate for initiated connections also specifies the version of SSL/TLS to support. AA-SBC supports SSL versions 2 and 3, and TLS version 1. You can specify any combination of these but be certain that at least one is set to **true** or AA-SBC will not be able to make connections. (By default, SSLv3 and TLSv1 are set to **true**.)

Default vs. specific CA and CRL files

AA-SBC can apply both a Certificate Authority (CA) and Certificate Revocation List (CRL) to a certificate. The CA file contains certificates that a peer can use to validate the certificate AA-SBC presents to it during an SSL/TLS connection (see default-ca for more information). The CRL file is a list of certificates that a CA has revoked, and thus can no longer be trusted (see default-crl for more information). In each case, you have the option to configure a default file and/or a specific file to the certificate.

- **default**—a file, or set of files, that are applied to all certificates. AA-SBC applies the files to a certificate if the **use-default-ca**/**use-default-crl** property is set to true. If set to false for a particular certificate, then that certificate does not use the default files. (You can still configure the certificate to use a CA or CRL file using the specific option, described below.)
- **specific**—a specific file that is applied either in addition to (if use-default is set to true) or instead of (if default is set to false) the default CA or CRL files for an individual certificate. You specify the specific file with the **specific-ca-file** property of this or the certificate objects.

Syntax

```
config vsp tls default-outgoing-settings
```

Properties

Property name	Description
certificate-file *filePath*	Specifies the name of the certificate file used to establish connections made with this object. The system supports the following certificate formats: • **PKCS#12**—Public Key Cryptography Standard #12 format, often from Microsoft IIS Version 5 (binary) • **PEM**—Privacy Enhanced Mail format, from any OpenSSL-based web server (ASCII) Enter a full path name to identify the certificate location. **Example: set certificate-file /cxc/certs/example.p12** There is no default setting.
passphrase-tag *string*	Specifies the passphrase associated with the certificate file. Use this property if the certificate file is encrypted to have its private key information protected. This passphrase must match the string that the certificate was encrypted with. **Example: set passphrase-tag abc123xyz** There is no default setting.

Property name	Description
allow-sslv2 {true | false}	Specifies whether the system can negotiate Secure Socket Layer Version 2 sessions with a peer. By default, the system only supports SSLv3 or TLSv1. If you require SSLv2 for interoperability, set this property to **true**. Note that SSLv2 is considered to suffer serious security holes. **Example: set sslv2 true** The default setting is **false**.
allow-sslv3 {true | false}	Specifies whether the system can negotiate Secure Socket Layer Version 3 sessions with a peer. **Example: set sslv3 true** The default setting is **true**.
allow-tlsv1 {true | false}	Specifies whether the system can negotiate Transport Layer Security Version 1 sessions with a peer. (TLSv1 is the IETF-approved version of SSLv3.) **Example: set tlsv1 true** The default setting is **true**.
allow-null-cipher {enabled | disabled}	Specifies whether to use a null string in the client Hello. This setting is ignored if you have a value set in the cipher-config-string property. Note that this property should never be enabled in a production environment, as it disables encryption. It is for debugging purposes only. If you do enable the null cipher, the client must list only the null cipher in its client Hello. Because the null cipher disables encryption, if any alternative is listed, the server will use it. **Example: set allow-null-cipher enabled** The default setting is **disabled**.

Property name	Description
dynamic-buffers {enabled | disabled}	Specifies whether to use dynamic buffers, an enhancement to the OpenSSL library. When **enabled**, the system allocates and frees transmit and receive buffers as they are needed, allowing support for many more TLS connections. When **disabled**, the system allocates both a transmit and a receive buffer at connection time, and holds the buffers open until the connection is dropped. **Example: set dynamic-buffers enabled** The default setting is **enabled**.
enable-cbc-counter-measure {true | false}	Enables or disables an OpenSSL strategy the system uses when sending TLS records. The strategy is designed to prevent an attack on cipher block chaining (CBC) ciphers, which have a vulnerability in some SSL implementations. If set to **false**, you disable the protection. **Example: set enable-cbc-counter-measure false** The default setting is **true**.
tx-record-length *bytes*	Sets the record length for TLS packets. By setting the length to a value less than the default of 16,384 bytes, you reduce the amount of memory required on transmit. **Example: set tx-record-length 4096** Enter a value from 1024 to 16,384. The default record length is **2048**.

Property name	Description
peer-certificate-verification {none | if-presented | required}	Specifies whether the system requests a certificate from a peer and the action in takes in response to the peer response. Select: • **none**—the system does not request a certificate from a peer. The system allows the connection whether or not peer presents a certificate. (If the peer does present, the system ignores the certificate.) • **if-presented**—the system requests a certificate from the peer. If the peer presents, the certificate must pass verification for the connection to proceed. If the peer does not present, the system allows the connection. Use this setting only when the system functions as a server. See Certificate presentation for details. • **required**—the system requests a certificate from the peer. If no certificate is presented, or if the presented certificate does not pass verification, the system terminates the connection. **Example: set peer-certificate-verification required** The default setting is **none**.
use-default-ca {true | false}	Specifies whether to use the default revocation list(s) configured in the default-ca object. **Example: set use-default-ca false** The default setting is **true**.
specific-ca-file *filePath passwordTag*	Specifies a CA file, and optionally a password, that should be used in addition to, or instead of, the default CA file(s). See Default vs. specific CA and CRL files for more information. **Example: set specific-ca-file cxc/certs/caZ.pem tagA** There is no default setting.

Property name	Description
use-default-crl {true | false}	Specifies whether to use the default revocation list(s) configured in the default-crl object. **Example: set use-default-ca false** The default setting is **true**.
specific-crl-file *filePath passwordTag*	Specifies a CRL file, and optionally a password, that should be used in addition to, or instead of, the default CA file(s). See Default vs. specific CA and CRL files for more information. **Example: set specific-crl-file cxc/certs/crl99.pem tag1** There is no default setting.
required-peer-name *string*	Specifies a name that must appear in the presented certificate. If you do not set this property, the system does not check the presented name. If you do specify a name, then it must appear in either the DNS field of the altSubjectName field or in the Common Name field. To verify the peer, the system first checks to see whether there is an entry in the DNS field of the altSubjectName field. If there is, the system compares it to the required-peer-name. If it matches, the system allows the connection (and performs no further peer-name checks). If the names do not match, the system disallows the connection (and performs no further peer-name checks). If there is no entry in the DNS field, then the system checks the Common Name field. If there is a match, the system allows the connection. If the presented name does not match the required name, the system rejects the connection. You can use wildcards to express the name. **Example: set required-peer-name *.companyABC.com** There is no default setting.
cipher-config-string *string*	*Secondary property.* Sets ciphers using the OpenSSL method. Do not change this parameter unless instructed to by Technical Support personnel.

default-ca

Purpose

Creates a pointer to a Certificate Authority file, which then becomes the default CA file for all certificates. This file contains certificates that a peer can use to validate the certificate AA-SBC presents to it during an SSL/TLS connection. See Default vs. specific CA and CRL files for information on using the default CA file.

CA files can be stored anywhere. However, if you store them in /cxc/certs directory, AA-SBC copies them to all systems in the cluster. To make a CA file available for use, simply copy it on to AA-SBC.

Syntax

```
config vsp tls default-ca
```

Properties

Property name	Description
ca-file *path* [*passwordTag*]	Specifies the path to the CA file and the password (if required) to access the file. The default-outgoing-settings and certificate objects can then be configured to use or ignore these default files. You can specify multiple CA files to be used as part of the default. **Example: set ca-file /cxc/certs/ca1.pem tag1** There is no default setting.

default-crl

Purpose

Creates a pointer to the one or more Certificate Revocation List (CRLs), which AA-SBC uses for certificate validation. The CRL file is a list of certificates that a CA has revoked, and thus can no longer be trusted. If any of the certificates in the chain presented to AA-SBC appear in the CRL, then AA-SBC rejects the connection.

Each certificate authority maintains an up-to-date list of revoked certificates. Because there are multiple CAs, you may wish to use multiple CRLs. CRL files can be stored anywhere. However, if you store them in /cxc/certs directory, AA-SBC copies them to all systems in the cluster. To make a CRL file available for use, simply copy it on to AA-SBC.

See Default vs. specific CA and CRL files for information on using the default CRL file.

Syntax

```
config vsp tls default-crl
```

Properties

Property name	Description
crl-file *path* [*passwordTag*]	Specifies a path to a CRL file and a password for that file. The default-outgoing-settings and certificate objects can then be configured to use or ignore these default files. You can specify multiple CRL files to be used as part of the default. **Example: set crl-file /cxc/certs/crl1.pem tag3** There is no default setting.

certificate

Purpose

Creates a certificate object that sets characteristics of certificate use. The object is then used in other areas of the configuration to associate a certificate with a TLS connection. See Using certificate vs. default-outgoing-settings for more information.

Enter a certificate name to open this object.

Syntax

```
config vsp tls certificate name
```

Properties

See the default-outgoing-settings object.

73. To and from interface group objects

Interface group description

The to- and from-interface group objects create groups for inbound and outbound traffic to which you can then apply traffic policy. See Chapter 48, “Policy objects” for a reference on traffic policy objects. For more information on creating AA-SBC policies, refer to the *Net-Net OS-E – Session Services Configuration Guide*.

Interface group object summary

The following table lists and briefly describes the **to-interface-group** and **from-interface-group** objects. See the following chapter for other objects in the CLI hierarchy:

- Chapter 75, “Virtual system partition (VSP) objects”

Object name	Description
`to-interface-group`	Sets policy for outgoing traffic based on destination(s) and/or egress interface(s).
`from-interface-group`	Sets policy for incoming traffic based on source(s) and/or ingress interface(s).

to-interface-group

Purpose

Creates or edits a group to which outgoing traffic policy is applied. AA-SBC assigns policy to outgoing traffic based on the destination network(s) and/or egress interface(s).

Syntax

```
config vsp to-interface-group name
```

Properties

Property name	Description
ip-network *ipAddress/mask*	Identifies the destination network for traffic belonging to this group. If packets are destined to the specified network, the system applies the named policy. Enter one or more IP address and mask combinations using CIDR notation. **Example: set ip-network 10.10.0.0/24** There is no default setting.

Property name	Description
interface *interfaceReference*	Sets the interface(s) that belong to the group. The system applies the specified policy to traffic sent out over the named interface(s). Enter the path to one or more previously configured interfaces. **Example: set interface cluster box 1 interface eth0 ip a** There is no default setting.
policy *`policyReference`*	Sets the policy that the system applies to all traffic: • destined for the network(s) identified in the **ip-network** property, and/or • sent out over the interface(s) identified in the **interface** property. Enter the path to a previously configured policy. **Example: set policy vsp policies session-policies policy default** There is no default setting.

from-interface-group

Purpose

Creates or edits a group to which incoming traffic policy is applied. AA-SBC assigns policy to incoming traffic based on the source network(s) and/or ingress interface(s).

Syntax

```
config vsp from-interface-group name
```

Properties

Property name	Description
ip-network *ipAddress*/*mask*	Identifies the source network of traffic belonging to this group. If packets originated from the specified network, the system applies the named policy. Enter one or more IP address and mask combinations using CIDR notation. **Example: set ip-network 192.168.0.0/24** There is no default setting.
interface *interfaceReference*	Sets the interface(s) that belong to the group. The system applies the specified policy to traffic arriving on the named interface(s). Enter the path to one or more previously configured interfaces. **Example: set interface cluster box 1 interface eth0 ip a** There is no default setting.
policy *policyReference*	Sets the policy that the system applies to all traffic: • destined for the network(s) identified in the **ip-network** property, and/or • sent out over the interface(s) identified in the **interface** property. Enter the path to a previously configured policy. **Example: set policy vsp policies session-policies policy default** There is no default setting.

74. User objects

User description

The user configuration object allows you to define in your configuration which users can pass SIP traffic on this VSP. This feature is only used if your SIP configuration requires local authentication. (Local authentication is set either in the **default-session-configuration** object under VSP, or the **session-configuration** object under policy/rule.) This object can also be used to authenticate users when stun-server **authentication-mode** is set to **local** (used for long-term authentication).

When you enable the local authentication file, you configure AA-SBC to prompt users that are passing SIP traffic to log in. The user name and password they enter must match the entries in this file. However, you can also create policy that, for example, does not attempt to authenticate users listed in the Active Directory.

User object summary

The following table lists and briefly describes the **user** objects and properties. See the following chapter for other objects in the CLI hierarchy:

- Chapter 75, “Virtual system partition (VSP) objects”.

Object name	Description
user	Configures names and passwords for users passing SIP traffic through the VSP.

user

Purpose

Configures user access to participation in SIP traffic on this VSP by adding an entry to the local authentication file. You enter each user individually, and assign a password. Use the admin property to allow or deny permission to pass SIP traffic. This file is only used if authentication is set to local in the **default-session-configuration** or **session-configuration** objects.

Enter the name of the user, up to 32 alphanumeric characters, to open the object. If the string contains delimiters (white space or \ character), it must be enclosed in double quotes (for example, “user name“). The name (and in the next level, password) that you configure are the logins needed by the user when AA-SBC prompts.

Syntax

```
config vsp user name
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables ability to pass SIP traffic on the VSP. You can use this property to temporarily disable a user without removing the user entry from the configuration. **Example: set admin disabled** The default setting is **enabled**.
password-tag *string*	Specifies the tag associated with the shared secret used to authenticate transactions between the system and this VSP user. See Using passwords and tags for information on the AA-SBC two-part password mechanism. **Example: set password abc124xyz** There is no default setting.

75. Virtual system partition (VSP) objects

VSP description

The AA-SBC virtual system partition (VSP) is the part of the system that holds the comprehensive customer-defined configuration, which controls how AA-SBC processes, stores, directs, and routes SIP traffic. The VSP is where you create session configurations, registration and dial plans, and policies that handle SIP request and response messages (for example, REGISTER requests and INVITE traffic). Based on the components of the VSP configuration, AA-SBC forwards traffic to SIP call destinations, authentication and accounting databases, VoIP service providers or carriers, enterprise servers, and so on.

VSP object summary

The following table lists and briefly describes the **vsp** objects. Only the vsp object is described in detail in this chapter. For all other objects (subobjects of **vsp**) full object descriptions can be found in the chapter referenced in the table,

Object name	Description
`vsp`	Sets the properties of the virtual system partition.
`displayname-character-set-info`	Maps character set transliteration options for user agents. Refer to Chapter 24, “Display name character set objects” for information.
`access`	Defines the users that are allowed access to AA-SBC and the specific privileges that they are granted. Refer to Chapter 5, “Access objects” for information.

Object name	Description
`default-session-config`	Specifies the default configuration used for all sessions on this VSP before policy is applied. Refer to Chapter 18, “Default session configuration objects” for information on setting the pre-policy configuration.
`autonomous-ip`	Configures groups and private gateways to allow AA-SBC to perform smart anchoring. Refer to Chapter 9, “Autonomous IP object” for information on configuring smart anchoring.
`tls`	Configures certificate and private key information. Refer to Chapter 72, “TLS objects” for information.
`pre-session-config`	Sets the parameters used by the VSP to alter SIP traffic before a session is established. Refer to Chapter 51, “Pre-session configuration objects” for information on setting the pre-session SIP parameters.
`policies`	Configures policies for this VSP, enterprise servers, and users. Refer to Chapter 48, “Policy objects” for information on setting AA-SBC policy.
`user`	Configures names and passwords for users passing SIP traffic through the VSP. Refer to Chapter 74, “User objects” for information.
`static-stack-settings`	Sets a variety of properties that can not be set dynamically. Refer to Chapter 67, “Static stack settings objects” for information.

Object name	Description
`virtual-threads`	Sets queue depth for the processing that relies on or is impacted by the location cache. Refer to Chapter 76, “Virtual threads objects” for information.
`session-config-pool`	Creates a session configuration that can be referenced through a dial plan. Refer to Chapter 63, “Session configuration pool objects” for information.
`dial-plan`	Configures a plan that directs outgoing phone calls to a particular gateway based on the dial prefix or domain suffix. Refer to Chapter 21, “Dial plan objects” for more information.
`registration-plan`	Allows you to delegate, proxy, forward, or redirect a registration request to a particular gateway based on the dial prefix or domain suffix. Refer to Chapter 55, “Registration plan objects” for more information.
`enterprise`	Configures access to the required databases to derive the recognized SIP addresses within your enterprise. Refer to Chapter 27, “Enterprise objects” for information on configuring enterprise gateways.
`carriers`	Configures the elements necessary to execute routing arbitration in an environment with multiple gateways. Refer to Chapter 12, “Carriers objects” for information on configuring carriers, gateways, and trunks.

Object name	Description
`calling-groups-`	Creates different groups (profiles) that can be referenced through the registration plan route and source-route configurations. Refer to Chapter 11, “Calling Group objects” for information.
`accounting`	Configures RADIUS and Diameter accounting services, system logging (syslog), the accounting database, and the accounting file-system. Refer to Chapter 6, “Accounting objects” for information.
`monitor-group`	Configures an endpoint, other than the call recipient, that can listen in on a call. Refer to Chapter 42, “Monitor-group objects” for information.
`radius-group`	Configures the authentication and accounting services associated with a group of RADIUS servers. Refer to Chapter 54, “RADIUS-group objects” for information.
`diameter-group`	Configures AA-SBC to operate as a Diameter client or server, providing AAA, 3Gpp Rx, and route server services. Refer to Chapter 22, “Diameter client and server objects” for information.
`im-filtering`	Creates lists to of words and URLs to scan against instant message content. Refer to Chapter 34, “IM Filtering objects” for information.
`from-interface-group`	Creates groups for inbound traffic to which you can then apply traffic policy. Refer to Chapter 73, “To and from interface group objects” for information.

Object name	Description
`to-interface-group`	Creates groups for outbound traffic to which you can then apply traffic policy. Refer to Chapter 73, “To and from interface group objects” for information.
`dns`	Configures AA-SBC to function as a DNS client (resolver). Refer to Chapter 25, “DNS service resolver and server objects” for information.
`location-service`	Configures the location service database for storing SIP caller location (or domain) information. Refer to Chapter 38, “Location service objects” for information.
`phones`	Sets AA-SBC to allow replication of phone configuration for supported models. Refer to Chapter 47, “Phone objects” for information on VoIP phone setup.
`presence-database`	Enables the AA-SBC presence database scan, which maintains necessary information for call forking. Refer to Chapter 50, “Presence database objects” for information.
`database`	Sets the number of days of records and archives that should be saved when AA-SBC performs database maintenance. Refer to Chapter 17, “Database object” for information.
`detect`	Helps AA-SBC to recognize the user agent (UA) sending a SIP message through the box. Refer to Chapter 20, “Detect objects” for information.

Object name	Description
`settings`	Sets a variety of advanced VSP parameters (which do not typically need modification). Refer to Chapter 64, “Settings objects” for information.
`admission-control`	Sets thresholds which determine the rate at which traffic is accepted onto the box. Refer to Chapter 7, “Admission Control objects” for information.
`services-routing`	Applies cluster-wide routing configurations, such as metrics to each of the AA-SBC service routing tables. Refer to Chapter 60, “Services routing objects” for information.
`oci-settings`	Configures the interface on AA-SBC that interacts with the BroadWorks server. Refer to Chapter 45, “OCI settings objects” for information.
`authentication`	Configures the authentication cache, which provides a mechanism for AA-SBC to throttle registrations that need authentication in the event that a slow RADIUS server impedes processing. Refer to Chapter 8, “Authentication and authorization objects” for information.
`dtmf-generation`	Controls the length of play and pause time and volume for the conference code digits that AA-SBC plays to a conference server on behalf of a client. Refer to Chapter 26, “DTMF generation objects” for information.
`registration-service`	*Secondary object*. Enables and sets expiration times for a registration service. Refer to Chapter 56, “Registration service objects” for information.

Object name	Description
sip-manipulation	
`sip-manipulation-pool`	

vsp

Purpose

Opens the virtual system partition (VSP) configuration object for editing. From this object you access the subobjects that set policy, session configuration characteristics, servers and directories, and many other aspects of system operation.

The properties of the VSP object include settings that limit sockets, sessions, SIP header and message sizes, and timers. You can also set the call admission control features within this object. The settings object contains a variety of advanced VSP parameters, which do not typically need modification.

Several properties within this object can be configured to allow AA-SBC to determine the appropriate value (a setting of **automatic**). See Using automatic values for more information.

Syntax

```
config vsp
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables this virtual system partition on this system. **Example: set admin enabled** The default setting is **disabled**.
local-identity *URL*	Sets the address-of-record for this SIP proxy. Use this property when there is more than one system (or multiple interfaces on a single system) sharing a location cache. The system writes the value of the local identifier field to the From header of a REGISTER request. **Example: set local-identity sip:boston@companyABC.com** There is no default setting.
local-normalization {enabled | disabled}	Specifies whether the system changes the host portion of the SIP header to the VSP domain name. If **enabled**, when a SIP request comes in and the host portion matches the system interface address or domain alias, the system changes the Host header. If **disabled**, the Host remains as it was originally received. **Example: set local-normalization disabled** The default setting is **disabled**.

Property name	Description
server-normalization {enabled | disabled}	Specifies a first level of control that dictates whether the system changes the Host portion of the SIP header to the matching server domain name. If **enabled**, and if the matching server has the normalization routing attribute enabled, the system changes the Host header when it receives a SIP request and the Host portion matches the system interface address or domain alias. If **disabled**, the Host remains as it was originally received. If your configuration includes many servers, you should set server-normalization to **disabled** to protect performance. **Example: set server-normalization disabled** The default setting is **disabled**.
external-inbound-normalization {no | yes *tableTag*}	Specifies whether the system should perform external normalization on inbound call legs from endpoints. If set to **no**, the system does not apply normalization settings. If set to **yes**, you must supply a tag to match against that determines the endpoints to which the system applies normalization. This name must match the table-tag column from the normalization.xml file **Example: set external-inbound-normalization yes endpnt** The default setting is **no**.

Property name	Description
registration-proxy {enabled | disabled}	Specifies how the system handles registrations on this VSP. If **enabled**, for each AOR in its location cache, if there is a matching registration plan to an upstream server and the upstream server has registration-proxy (see below) configured and enabled, the system originates registrations to the upstream server. If **disabled**, the system does not originate a registration for AORs in its location cache. You can also set registration-proxy characteristics for a server, which allows peers to receive proxy information. When set at the VSP level (this property), you are optimizing performance for your system. **Example: set registration-proxy enabled** The default setting is **disabled**.
pstn-gateway {none | server *serverReference* | carrier *carrierReference* | gateway *gatewayReference* | trunk *trunkGrpReference* | hunt-group *huntGrpReference*}	Identifies the configured PSTN gateway for an enterprise server with a PSTN backup configuration. (This is set with the server routing setting property.) The system can allow enterprises to continue call operations even if their provider server is busy or down. You do this by configuring the public switched telephone network (PSTN) gateway. Normally, the system forwards calls to a provider application server. If the server has failed, and the system has location information for the endpoints, it forwards calls locally. Otherwise, if configured, the system forwards calls to the PSTN gateway you configured with the sip-gateway server object. Enter a path to that configured server. **Example: set pstn-gateway vsp enterprise servers** **sip-gateway pstn** The default type is **server**, with no default server reference.

Property name	Description
external-policy-group *serverReference*	References a policy server configuration. That configuration sets the URL of the external server that maintains policy configurations to apply to a session. See policy-service for more information. **Example: set external-policy-group "external-services policy-group myPolicy"** There is no default setting.
external-location-group *serverReference*	References a location service configuration. That configuration sets the URL of the VoIP Positioning Center (VPC) providing location services (caller location) for VoIP subscribers using Location Information Services (LIS). See red-sky-location-service, tcs-location-service, or generic-service for more information. **Example: set external-location-group "external-services location-group forE911"** There is no default setting.
external-event-group *serverReference*	References an event server configuration. That configuration sets the URL of a server used for tracking system events. (These events are similar to SNMP traps.) See event-service for more information. **Example: set external-event-group "external-services external-group myEvents"** There is no default setting.

sip-manipulation

Purpose

Configure a specific SIP manipulation. This lets you add, modify, and delete SIP headers and parts of the SIP headers called SIP header elements. SIP header elements are the different subparts of the header, such as the header value, header parameter, and URI parameter.

Syntax

```
config vsp sip-manipulation-pool sip-manipulation
```

Properties

Property name	Description
description	Description of this SIP manipulation object. **Example: set description manip10092009**
header-rule	Enter a list of header rules for this SIP manipulation. **Example: set header-rule delete**

sip-manipulation-pool

Purpose

Secondary object. Allows you to configure the pool of named SIP manipulation objects for the AA-SBC. These contain lists of SIP header manipulation rules and elements.

Syntax

```
config vsp sip-manipulation-pool
```

Properties

Property name	Description
sip-manipulation	Enter the SIP manipulation you want to configure. EXAMPLE: **set sip-manipulation manip1**

sip-response-q931-cause-map

Purpose

Allows the configuration of sip-response code for calls cleared by an external H.323 GW. When a ReleaseComplete, Admission Reject, or Location Reject message is received by the AA-SBC, the AA-SBC consults an internal table to determine the appropriate SIP response code to generate when clearing the SIP side of the call. By adding a **sip-response-q931-cause-map** entry, you can override the internal table defaults.

Syntax

```
config vsp enterprise servers h323-server sip-response-q931-cause-map
```

Properties

Property name	Description
sip-response	Define the sip-response that will be used when clearing the SIP side of the call. Example: set sip-response 350 Min: 300 / Max: 699 The default setting is **0**.
q931-cause	Select a q931-cause that helps to qualify the H.323 call-clear. If this map entry does not depend on the q931-cause value, either because there is no Q.931 present or because any Q.931 cause qualifies, choose **Any**. Example: **set q931-cause noresponse**
h2250-reason	Select a h2250-reason type. The following are valid values: • LRJ—match an incoming LRJ message. • ARJ—match an incoming ARJ message. • H.225—match all other relevant traffic • none—H.225 reason should not be used as match criteria for this entry. Example: set h225-reason lrj The default setting is **none**.

76. Virtual threads objects

Virtual threads description

Sets the depth of the queues used by AA-SBC when determining processing availability for aging the location cache. See Chapter 38, “Location service objects”, for more information on location cache aging. Specifically, see the settings object **max-cache-poll-duration** property for more information.

Virtual threads object summary

The following table lists and briefly describes the **virtual-threads** object. See the following chapter for other objects in the CLI hierarchy:

- Chapter 75, “Virtual system partition (VSP) objects”

Object name	Description
`virtual-threads`	Sets queue depth for the processing that relies on or is impacted by the location cache.

virtual-threads

Purpose

Specifies the depth of the various queues used for processing SIP messages. AA-SBC uses these queue levels to manage purging of the location cache. See the location cache settings object **max-cache-poll-duration** property for more information.

Syntax

```
config vsp virtual-threads
```

Properties

Property name	Description
urgent-congestion-threshold *threads*	Sets the threshold for the urgent queue. The system handles all internal signaling messages through the urgent queue. **Example: set urgent-congestion-threshold 192** The default setting is **128** threads.
priority-congestion-threshold *threads*	Sets the threshold for the priority queue. The system handles all SIP messages through the priority queue. **Example: set priority-congestion-threshold 384** The default setting is **256** threads.
regular-congestion-threshold *threads*	Sets the threshold for the regular queue. The system receives the timeout event from the location cache aging timer into this queue. Or, if aging is not complete, the event indicating the next group of messages can be aged. **Example: set regular-congestion-threshold 16** The default setting is **32** threads.

77. VLAN objects

VLAN description

A virtual local area network (VLAN) is a logical grouping of systems that is not constrained by geographic boundaries. These groupings create a broadcast domain, and function just like a traditional LAN. Systems within the VLAN are not necessarily physically co-located, but do not require a router to connect them. (Routers are used to connect separate VLANs). VLANs are interconnected using system bridging software.

VLAN tagging

AA-SBC supports VLAN tagging, allowing a 802.1Q-compliant VLAN identifier to be added to the packet before it is sent. VLAN tagging (i.e., whether a VLAN tag is added to the packet on transmit) is enabled per VLAN, on a per interface basis. When a packet is received with a VLAN tag, the packet is accepted if the receive port is an active member of the tagged VLAN.

AA-SBC implements its virtual firewall functionality through the use of VLAN tagging to partition private network segments. For information, see the *Net-Net OS-E – System Administration Guide*.

VLAN object summary

The following table lists and briefly describes the **vlan** objects. The VLAN configuration object is contained in the cluster and box configuration hierarchies. See the following chapter for other objects in the CLI hierarchy:

- Chapter 14, “Cluster, box, and interface objects”

Object name	Description
`vlan`	Creates or edits a VLAN on an interface.

vlan

Purpose

Creates Layer 2 partitions, grouping LAN segments so that they appear to be on the same Layer 2 network. Each VLAN is identified by a VLAN ID; the ID must be unique within portions of the network, depending on the use of the VLAN.

If VRRP is not in use, a VLAN ID must be unique to each physical interface. A physical interface consists of the bottom-level interface (e.g., eth0) and all interfaces within a virtual firewall hosted on that bottom-level interface. VLAN IDs can overlap on different physical interfaces, however. For example, you could have VLAN 20 assigned on both interface eth0 and eth1.

If a cluster is configured with a VRRP interface, VLANs on the cluster must be unique among all interfaces on the cluster. (This includes the VLANs that you configure in a virtual firewall on the cluster.)

Enter a value between 2 and 4095. AA-SBC creates a new VLAN with the specified number as an ID or opens for editing an existing VLAN.

Syntax

On a public IP interface:

```
config box interface ethX vlan integer
config cluster box integer interface ethX vlan integer
config cluster vrrp vinterface vxID vlan integer
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables VLAN services on the specified Ethernet interface. **Example: set admin enabled** The default setting is **enabled**.
arp {enabled | disabled}	Enables or disables Address Resolution Protocol (ARP) services on this VLAN interface. ARP is the Internet protocol that maps IP addresses to corresponding Ethernet addresses. **Example: set arp enabled** The default setting is **enabled**.

78. VRRP objects

VRRP description

The VRRP objects allow you to configure the Virtual Router Redundancy Protocol (VRRP) on AA-SBC. VRRP provides link-level failover capabilities and continued service between two or more (and up to 255) AA-SBC virtual interfaces should a network link go down in the network cluster.

Within VRRP each interface is assigned a master or backup responsibility. The priority of an interface is determined by the order in which you added the interface as a host (i.e., the first interface added has the highest priority).The VRRP interface with the highest priority is responsible for forwarding traffic and is the elected master. Other configured VRRP interfaces across the cluster with lower priorities serve as VRRP backup interfaces available to assume VRRP mastership, if necessary. You can change the order of the interfaces using the move command.

If the master VRRP interface becomes unavailable, the election protocol enables a backup VRRP interface to assume mastership using the next prioritized (added) interface. However, if the original master VRRP interface (the interface with the highest priority) should once again become available (and the preempt property is set to true), VRRP returns mastership to that interface.

Note that in addition to configuring VRRP for the cluster, you must also enable VRRP on one IP interface. Make sure that the parent physical interface is one that is always available, such as the management interface.

See *RFC 2338, Virtual Router Redundancy Protocol*, for detailed information about this protocol.

VRRP object summary

The following table lists and briefly describes the **vrrp** objects.

Object name	Description
cluster: vrrp	Opens the vrrp configuration object for editing.
vinterface	Configures a virtual router interface to participate in VRRP.
ip	See Chapter 35, “IP objects”, for information.
vlan	See Chapter 77, “VLAN objects”, for information.
interface: vrrp-advertisements	Specifies the interface over which AA-SBC sends out VRRP advertisements.

vrrp

Purpose

Sets the administrative status of VRRP on a cluster. On AA-SBC, VRRP functions as virtual Ethernet redundancy protocol, setting up failover for Ethernet interfaces.

In order for VRRP to work effectively on a cluster, you must enable the **share-registration-entries** property in the cluster object.

Syntax

```
config cluster vrrp
```

Properties

Property name	Description
fault-group *groupNumber*	*Secondary property.* Sets the system to initiate a VRRP failover to the backup box if there is a SIP crash. When you specify a group, the system associates the SIP process with that group. (It is a good idea to apply this group to interfaces carrying SIP traffic.) In the event of a crash, the system brings down the referenced VRRP group, causing the failover. VRRP groups are configured using the vinterface **group** property. A value of 0 disables this function. **Example: set fault-group 5** The default setting is **0**.
media-fault-group *groupNumber*	*Secondary property.* Sets the system to initiate a VRRP failover to the backup box if there is a media crash. When you specify a group, the system associates the SIP process with that group. (It is a good idea to apply this group to interfaces carrying SIP traffic.) In the event of a crash, the system brings down the referenced VRRP group, causing the failover. VRRP groups are configured using the vinterface **group** property. A value of 0 disables this function. **Example: set media-fault-group 3** The default setting is **0**.

vinterface

Purpose

Configures a virtual interface in the cluster, for use by VRRP. A vinterface is a VRRP construction that allows a backup system for Ethernet interfaces. It does so by updating and advertising MAC-to-IP address mappings (using gratuitous ARP). A VRRP virtual router defines a MAC address for an interface. AA-SBC loads that address onto the Ethernet interface acting as the master host. If that interface goes down, AA-SBC moves that MAC address to the next priority interface, using ARP to broadcast the MAC-to-IP mapping.

To create or edit a vinterface, enter a virtual router interface ID in the range of **vx0** to **vx254**.

Syntax

```
config cluster vrrp vinterface vxID
```

Properties

Property name	Description
admin {enabled | disabled}	Enables and disables this VRRP virtual interface configuration. When **enabled**, the referenced interfaces participate in the failover features of VRRP. When **disabled**, the interfaces do not serve as link backup. **Example: set admin disabled** The default setting is **enabled**.
group *groupID*	Groups VRRP interfaces for failover purposes. If you assign a vinterface a group number of 0, it does not participate in grouping. Services that run on the box can use the VRRP grouping feature for failover as well. See Master services in VRRP configurations for more information. **Example: set group 10** Enter a value for the VRRP group. The default setting is 0 (no grouping association).

Property name	Description
host-interface *interfaceReference*	Sets the Ethernet interfaces to be used as part of the VRRP resources. Re-execute the command for each interface in the VRRP pool. The order of priority is established by your order of entry (first is highest). You can configure two or more Ethernet interfaces on a box, but the boxes must be grouped. For example, all box 2 instances must occur in a row—you could configure box 2, box 2, box 1, box 3. A box must fail fully (all interfaces) before the system seeks connectivity with the next box in the configuration. Note that each Ethernet interface should only be used once, as a master or backup. Do not re-use interfaces across a vinterface. **Example: set host-interface cluster box 1 interface eth0** Enter a reference to a previously configured Ethernet interface. There is no default setting.
preempt {true | false}	*Secondary property.* Specifies whether the configured master vinterface should retake the mastership if it has gone down and then returned to operation. If set to **true**, the master resumes its position. If set to **false**, the backup interface retains master control. **Example: set preempt true** The default setting is **false**.
preempt-delay *seconds*	*Secondary property.* Specifies whether the system should wait for a period of time before restoring the master (if **preempt** is set to **true**). **Example: set preempt-delay 5** Enter a value from 0 to 120; the default setting is **0** (no delay).

Property name	Description
gratuitous-arp-count *integer*	*Secondary property.* Specifies the number of gratuitous ARP packets to send. The system uses gratuitous ARP to keep other devices informed of the IP-to-MAC address mapping of the current master vinterface. **Example: set gratuitous-arp-count 3** Enter a value from 1 to 20; the default setting is **1**.
gratuitous-arp-interval *seconds*	*Secondary property.* Specifies the number of seconds between gratuitous ARP packets if the count is set to greater than 1. **Example: set gratuitous-arp-interval 2** Enter a value from 1 to 20; the default setting is **1**.
gratuitous-arp-period *seconds*	*Secondary property.* Specifies the number of seconds the system waits between completing the previous gratuitous ARP count/interval, and beginning the next. For example, if you set the count to 2, the interval to 3, and the period to 10, the system sends out a GARP, waits 3 seconds and then sends another. After 10 seconds, the system begins again. A value of 0 sets the system to only send out the first alerting GARP packets. **Example: set gratuitous-arp-period 10** Enter a value from 0 to 65535; the default setting is **0**.
advertisement-timer-value *milliseconds*	*Secondary property.* Specifies how often the master VRRP interface advertises itself to other interfaces in the pool. Do not change this value unless instructed to do so by technical support. **Example: set advertisement-timer-value 200** The default setting is **100** milliseconds.

Property name	Description
heartbeat-timer-value *milliseconds*	*Secondary property.* Sets the basis of the value used by the system to determine how long to wait before failing over to the backup interface. Do not change this value unless instructed to do so by technical support. **Example: set heartbeat-timer-value 900** The default setting is **600** milliseconds.
takeover-skew *value*	*Secondary property.* Sets an internal value that influences the VRRP takeover timer calculation. Do not change this value unless instructed to do so by Technical Support. **Example: set takeover-skew 1** Enter a value between 1 and 3; the default setting is **1**.

vrrp-advertisements

Purpose

Sets the interface that AA-SBC uses to send out VRRP advertisements. Enable this object on one IP interface per box.

Syntax

```
config cluster box number interface ethX ip name vrrp-advertisements
config cluster box number interface ethX vlan number ip name
   vrrp-advertisements
config box interface ethX ip name vrrp-advertisements
config box interface ethX vlan number ip name vrrp-advertisements
```

Properties

Property name	Description
`admin {enabled | disabled}`	Sets the administrative state of VRRP advertisements on an interface. When **enabled**, the interface is used by the system to send out VRRP advertisements. When **disabled**, advertisements are not sent out over the interface. Enable this feature on only one interface per box. **Example:** set admin disabled The default setting is **enabled**.

79. Web objects

Web server description

The Web object enables the Web server, providing access to the AA-SBC Management System graphical user interface. If you want to view SNMP traps through the GUI, you must also enable the server as a trap target. You enable and configure Web services on Ethernet and VLAN interfaces.

Web object summary

The following table lists and briefly describes the **web** objects. See the following chapters for other objects in the CLI hierarchy:

- Chapter 14, “Cluster, box, and interface objects”
- Chapter 77, “VLAN objects”
- Chapter 35, “IP objects”

Object name	Description
`web`	Configures Web services on a AA-SBC interface.
`trusted-ips`	Adds clients to the list of trusted hosts and associates permissions.

web

Purpose

Configures the web server on an Ethernet or VLAN interface.

Syntax

```
config cluster box number interface ethX ip name web
config cluster box number interface ethX vlan number ip name web
config box interface ethX ip name web
config box interface ethX vlan number ip name web
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables the system web server configuration object. Enabling a web server allows you to manage the system using the AA-SBC Management System. **Example: set admin enabled** The default setting is **enabled**.
protocol {http *port* | https *port* [*redirectPort*] [*certificateReference*] [*alias*]	Sets the protocol to use for AA-SBC Management System operations. After setting a protocol, you can select the web server listening port (or accept the default). This is the port the server listens on for HTTP(S) requests. Select either: • **http**—sets an insecure (unencrypted) protocol for use in web transmission. • **https**—provides secure transmission of web pages by using HTTP over SSL. Optionally, you can set: • a redirect port • a reference to a previously configured certificate (configured with the certificate object) • an alias for the key in the key store (named in the certificate configuration). **Example: set protocol https 443 0 “vsp tls certificate nnos-e.company.com” certKey** The default protocol setting is **https** with a port setting of 443. The redirect port defaults to 0, which disables redirect. If you specify a certificate, the default alias for the key is tomcat. If you set the protocol to **http**, the default port is 80.

Property name	Description
trap-target {enabled | disable}	Configures whether the Web server collects SNMP traps. If **enabled**, the system sends traps to the Web server as well any other SNMP target. You can then view the SNMP traps through the AA-SBC Management System **Event Logs** tab. If **disabled**, the Web does not collect SNMP traps. **Example: set trap-target enabled** The default setting is **disabled**.
jmx {enabled [*registryPort*] [*serverPort*] | disabled}	Enables or disables the Java Management Extensions Managed Beans (MBeans) server. The MBean server functions as a management agent by acting as a registry for all manageable resources—applications, services, components, and devices. When **enabled**, the system uses JMX as an interface to control resources and make them available to remote management applications. Optionally, you can specify the When **disabled**, the server does not run and you use another method as a system interface. **Example: set jmx enabled** The default setting is **disabled**. If enabled, the default registry port is 1099 and the default server port is 1100.
max-threads *integer*	Specifies the maximum number of total worker threads, both active and spare (idle), allocated to the web server. **Example: set max-threads 15** Enter a value between 1 and 50; the default setting is **10** threads.
min-spare-threads *integer*	Specifies the minimum number of inactive threads that the system must leave allocated to the web server. When the system removes idle threads, it must leave this number of spares available. **Example: set min-spare-threads 3** Enter a value between 0 and 50; the default setting is **1** thread.

Property name	Description
max-spare-threads *integer*	Specifies the maximum number of inactive threads the system can leave allocated to the web server. When the system detects idle threads, it can not maintain more than this number. **Example: set max-spare-threads 8** Enter a value between 0 and 50; the default setting is **5** threads.
idle-timeout *minutes*	Specifies an inactivity timeout for the AA-SBC Management System. When a session has been inactive for this number of minutes, the system logs the user off the system. A value of 0 turns off the inactivity timer. **Example: set idle-timeout 45** The default setting is **30** minutes.

trusted-ips

Purpose

Configures a profile of one or more IP addresses that are considered “trusted hosts” and can therefore bypass authentication. When AA-SBC receives an HTTP request, it checks to determine whether the request came from an IP address within any trusted host profile. AA-SBC then applies the matching permissions to the request, and bypasses the authentication (login) process. Use this to integrate selected pages into a third-party application.

Syntax

```
config cluster box number interface ethX ip name web trusted-ips name
config cluster box number interface ethX vlan number ip name web
config box interface ethX ip name web trusted-ips name
config box interface ethX vlan number ip name web trusted-ips name
```

Properties

Property name	Description
permissions *permissionsReference*	Associates a set of permissions with this trusted host. Enter a reference to a previously configured set of permissions. Keep in mind that the specific access granted by the permissions profile effects the tab display in the AA-SBC Management System. **Example: set permissions “access permissions EMS”** There is no default setting.
ip-address *ipAddress*	Sets the IP address(es) that should be allowed access to the AA-SBC Management System without further authentication. Enter the IP address of the client accessing the Web browser. **Example: set ip-address 172.24.0.22** There is no default setting.

80. Web-service objects

Web services description

The web services object enables the Web Services Definition Language (WSDL). WSDL is an XML-based language for describing Web services, and how to access them, in a platform-independent manner. Simple Object Access Protocol SOAP (SOAP) is the communication protocol used for communication between applications, based on XML.

A WSDL document is a set of definitions that describe how to access a web service and what operations it will perform. AA-SBC uses it in combination with SOAP and XML Schema to allow a client program connecting to a web service to determine available server functions. The actions and data types required are embedded in the WSDL file, which then may be enclosed in a SOAP envelope. The SOAP protocol supports the exchange of XML-based messages, with AA-SBC using HTTPS.

You can configure AA-SBC as both a WSDL client and server. Use the external-services object to configure it as a client; use the web-service object to enable the interface, allowing AA-SBC to function as a server.

See *Net-Net OS-E – Using the NNOS-E Management Tools* for a complete description of the AA-SBC WSDL implementation.

Web services object summary

The following table lists and briefly describes the **web-services** objects. See the following chapters for other objects in the CLI hierarchy:

- Chapter 14, “Cluster, box, and interface objects”
- Chapter 77, “VLAN objects”
- Chapter 35, “IP objects”

Object name	Description
`web-service`	Configures AA-SBC as a Web services server.

web-service

Purpose

Configures AA-SBC as a web services server by enabling WSDL on the interface. Configuring AA-SBC as a server allows clients to make calls and requests into the box to control and manage the platform. To configure AA-SBC as a web service client, allowing it to make web service “call outs” to get location and policy information from an external service endpoint, use the external-services object.

Syntax

```
config cluster box number interface ethX ip name web-service
config cluster box number interface ethX vlan number ip name
   web-service
config box interface ethX ip name web-service
config box interface ethX vlan number ip name web-service
```

Properties

Property name	Description
admin {enabled | disabled}	Enables or disables the use of WSDL over the selected interface. If **enabled**, the system functions as a web services server. **Example: set admin enabled** The default setting is **enabled**.
protocol {http *port* | https *port* [*redirectPort*] [*certificateReference*] [*alias*]	Sets the protocol to use for AA-SBC Management System operations. After setting a protocol, you can select the web server listening port (or accept the default). This is the port the server listens on for HTTP(S) requests. Select either: • **http**—sets an insecure (unencrypted) protocol for use in web transmission. • **https**—provides secure transmission of web pages by using HTTP over SSL. Optionally, you can set: • a redirect port • a reference to a previously configured certificate (configured with the certificate object) • an alias for the key in the key store (named in the certificate configuration). **Example: set protocol https 443 0 “vsp tls certificate nnos-e.company.com” certKey** The default protocol setting is **http** with a port setting of 8080. If you set the protocol to **https**, the default port is 443. The redirect port defaults to 0, which disables redirect.

Property name	Description
authentication {basic | certificate *integer*}	Specifies whether the web service client needs a certificate to communicate with the system. Set the property to **basic** to require HTTP basic authentication for client connections. Set the property to **certificate** to require an HTTPS certificate for authentication of client connections. **Example: set authentication certificate "vsp tls certificate ws_cert"** The default setting is **basic**.
application *applicationReference*	Identifies a click-to-call profile that the system web service server will host. Enter a reference to a previously configured application. **Example: set application preferences click-to-call** There is no default setting.
max-threads *integer*	Specifies the maximum number of total worker threads, both active and spare (idle), allocated to the web service. **Example: set max-threads 15** Enter a value between 1 and 50; the default setting is **10** threads.
min-spare-threads *integer*	Specifies the minimum number of inactive threads that the system must leave allocated to the web service. When the system removes idle threads, it must leave this number of spares available. **Example: set min-spare-threads 3** Enter a value between 0 and 50; the default setting is **1** thread.
max-spare-threads *integer*	Specifies the maximum number of inactive threads the system can leave allocated to the web service. When the system detects idle threads, it can not have maintain more than this number. **Example: set max-spare-threads 8** Enter a value between 0 and 50; the default setting is **5** threads.

Property name	Description
max-message-process-threads	The maximum number of messaging processing threads. **Example: set max-message-process-threads** Min: 10 / Max: 200 The default setting is **10**.
max-http-connections	The maximum number of outbound http connections. **Example: set max-http-connections 250** Min: 100 / Max 300 The default setting is **100**.
max-http-client-connections	The maximum number of outbound HTTP connections per host. **Example: set max-http-client-connections 75** Min: 5 / Max: 100 The default setting is **10**.