Avaya Aura™ SBC System Administration Guide

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

Except where expressly stated otherwise, no use should be made of materials on this site, the Documentation(s) and Product(s) provided by Avaya. All content on this site, the documentation(s) and the product(s) provided by Avaya including the selection, arrangement and design of the content is owned either by Avaya or its licensors and is protected by copyright and other intellectual property laws including the sui generis rights relating to the protection of databases. You may not modify, copy, reproduce, republish, upload, post, transmit or distribute in any way any content, in whole or in part, including any code and software. Unauthorized reproduction, transmission, dissemination, storage, and or use without the express written consent of Avaya can be a criminal, as well as a civil, offense under the applicable law.

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.

Table of Contents

Preface

About AA-SBC documentation xiii
Conventions used in this manual xiv
Typographical conventions xiv
Acronyms xiv

1. Managing and administering AA-SBC systems

About this chapter 21
References 21
Administrator and user roles 21
Enabling management access 22
CLI session 22
Configuring management options 23
Local console 24
CLI session 24
Telnet 24
CLI session 24
Secure Shell (SSH) 25
CLI session 25
Web/HTTP 26
CLI session 26
SNMP 26
CLI session 27
HTTP\SOAP\WSDL interface 27
Working with the AA-SBC configuration file 28
Building the configuration file using the CLI 28
CLI session 29
Removing objects from the configuration file

using the CLI 29
CLI session 29
Editing and saving the configuration file using the CLI 30
Creating SIP users and passwords 30
CLI session 31
Customizing the CLI 31
CLI session 31
Setting AA-SBC global properties 32
CLI session 32
AA-SBC virtual system partitions 33
Specifying management preferences 33
Specifying DOS query preferences 34
Restarting and shutting down the system 35
CLI session 35

2. Enabling AA-SBC interfaces and protocols

About this chapter 37
AA-SBC sample networks 37
Configuring AA-SBC IP interfaces 39
CLI session for eth0 39
CLI session for eth1 40
CLI session for eth2 40
Creating VLANs 40
CLI session 41
Applying routing and classification tags 41
CLI sessions for “IP A” and “IP B” ingress networks on eth3 43
Notes on routing and classification tags 46
Related commands 46
Configuring overlapping IP networks and tag routing 47
CLI session for Ethernet public and private sides of network 47
CLI sessions for Customer-A and Customer-B networks 48
CLI session for the internal private network 49

CLI session for the session-config-pool 49
Configuring VRRP 50
CLI session 51
CLI session 51
Configuring signaling failover 54
CLI session 54
Configuring Web interface settings 55
CLI session 55
Configuring Web services 55
CLI session 55
Enabling ICMP and setting rate limits 56
CLI session 56
Enabling NTP and BOOTP servers 56
CLI session 57
Configuring the Bootstrap Protocol (BOOTP) clients 57
CLI session 58
Configuring SIP 58
CLI session 58
CLI session 59
Load balancing across AA-SBC interfaces 59
Configuring media port pools 60
CLI session 61
Configuring the external firewalls 61
CLI session 62
Configuring the STUN, TURN, and ICE protocols 63
STUN 63
Traversal Using Relay NAT (TURN) 64
Interactive Connectivity Establishment (ICE) 64
Sample configuration 65
CLI session 65
Configuring kernel filtering 66
CLI session 66
Configuring messaging 66
CLI session 67

3. Enabling AA-SBC services

About this chapter 69
Enabling services on the AA-SBC master 69
Cluster-master services 70
CLI session 70
Directory services 70
CLI session 70
Accounting services 70
CLI session 71
Authentication services 71
CLI session 71
AA-SBC database 71
CLI session 71
Registration services 72
CLI session 72
Server load 72
Call failover (signaling and media) 73
CLI session 73
Load-balancing 74
CLI session 74
File-mirror 75
Route Server 76
CLI Session 76
Sampling 77
CLI Session 78
Third-party-call-control (3PCC) 78
CLI Session 79
Enabling event logging services 79
CLI session 79
CLI session 80
CLI session 80
Configuring threshold monitors 80
Configuring data and archiving locations 81
CLI session 81

CLI session 82

CLI session 82

Configuring an external database 82

CLI session 83

Setting AA-SBC disk thresholds 83

CLI session 84

Scheduling regularly performed tasks 84

CLI session 84

Performing database maintenance 84

Setting normal database maintenance time-of-day 85

CLI session 85

Verifying normal database maintenance 85

Scheduling periodic database maintenance 86

CLI session 86

Forcing database maintenance 86

Performing database vacuum-full 87

Performing other database maintenance tasks 87

Managing Net-Net-OS-E database size 88

Disabling REGISTER message logging 89

Preventing NOTIFY message logging 90

Backing up the database 92

CLI session 93

CLI session 94

Restoring a database 94

Enabling and configuring local archiving 94

CLI session 95

CLI session 96

4. Configuring AA-SBC accounting and archiving

About this chapter 97

Accounting system overview 97

Configuring the accounting settings 98

Configuring RADIUS groups 99

CLI session 100

Configuring the RADIUS servers 101

CLI session 101
Including the RADIUS group 102
CLI session 102
Configuring the accounting database 103
CLI session 103
Configuring syslog 105
CLI session 105
Configuring the file system 107
CLI session 108
Configuring an External File System Target 108
CLI Session 109
Configuring Diameter 110
Creating the Diameter accounting group 110
CLI session 110
Configuring Diameter servers 111
CLI session 111
Configuring Diameter interfaces and ports 112
CLI session 112
Configuring archiving 113
CLI session 114
Using the AA-SBC archive viewer 116

5. Configuring Domain Name Systems (DNS)

About this chapter 119
Domain Name System (DNS) overview 119
Configuring the DNS resolver 120
CLI session 121
Configuring DNS hosts and IPs 121
CLI session 122
Mapping SIP services 122
CLI session 122
Configuring NAPTR 123
CLI session 123
Configuring DNS rejections 124
CLI session 124

6. Managing certificates and keys with LCS and OCS

About this chapter 125
Transport Layer Security overview 125
Acquiring certificates 125
Installing the CA chain file on LCS/OCS servers 126
Installing a certificate on LCS/OCS servers 131
Exporting certificates to AA-SBC 134
Importing certificates using file transfer 145
CLI session 145
Configuring the certificate on AA-SBC 146
CLI session 146
Managing cryptographic keys 147

7. Configuring SIP servers, directories, and federations

About this chapter 149
Servers, directories, and federations 149
Interoperating with SIP servers 150
Configuring the SIP server pools 150
IBM Lotus Sametime 151
CLI session 151
Microsoft LCS 151
CLI session 151
Nortel MCS 152
CLI session 152
Avaya PBX 153
CLI session 153
SIP hosts 153
CLI session 154
SIP gateways 155
CLI session 155
SIP registrars 156
CLI session 156
H.323 servers 157
SIP connections 157
DNS groups 158

Configuring directories 158

Active directory 159

CLI session 159

LDAP 159

CLI session 159

Notes directory 160

CLI session 160

Phantom directory 160

CLI session 160

Static directory description 161

CLI session 161

XML directory description 161

CLI session 161

Database directory description 162

CLI session 162

CSV directory description 162

CLI session 162

Configuring federations 163

Configuring 3PCC servers 163

Preface

About AA-SBC documentation

The following documentation set supports the current release of the Avaya Aura™ Session Border Controller (AA-SBC) software, powered by Acme Packet.

- *Avaya Aura™ SBC – Using the AA-SBC Management Tools*
- *Avaya Aura™ SBC – System Administration Guide*
- *Avaya Aura™ SBC – Session Services Configuration Guide*
- *Avaya Aura™ SBC – Objects and Properties Reference*
- *Avaya Aura™ SBC – System Operations and Troubleshooting*

Conventions used in this manual

Typographical conventions

Key Convention	Function	Example
KEY NAME	Identifies the name of a key to press.	Type **abc**, then press [ENTER]
CTRL+*x*	Indicates a control key combination.	Press CTRL+C
brackets [ ]	Indicates an optional argument.	`[`*`portNumber`*`]`
braces { }	Indicates a required argument with a choice of values; choose one.	`{enabled | disabled}`
vertical bar |	Separates parameter values. Same as "or."	`{TCP | TLS}`
Monospaced bold	In screen displays, indicates user input.	`config>` **`config vsp`**
Monospaced italic	In screen displays, indicates a variable—generic text for which you supply a value.	`config servers>` **`config lcs`** *`name`*
bold	In text, indicates literal names of commands, actions, objects, or properties.	...set as the secondary directory service (with the **unifier** property)...
bold italic	In text, indicates a variable.	...set the **domain** property of the ***directory*** object.

Acronyms

AA-SBC manuals contain the following industry-standard and product-specific acronyms:

AA-SBC	Avaya Aura™ Session Border Controller
AAA	Authentication, authorization, and accounting
ALI	Automatic location identifier
ANI	Automatic number identification
ANSI	American National Standards Institute
AOR	Address of record
API	Application programming interface
ARP	Address Resolution Protocol

AVERT	Anti-virus emergency response team
B2BUA	Back-to-back user agent
BOOTP	Bootstrap Protocol
CA	Certificate authority
CAP	Client application protocol
CBC	Cipher block chaining
CBN	Call back number
CCS	Converged Communication Server
CDR	Call detail record
CIDR	Classless interdomain routing
CLI	Command line interface
CMOS	Comparison mean opinion score
CNAME	Canonical name record
CNI	Calling number identification
CODEC	Compressor/decompressor or coder/decoder
CPE	Customer-premise equipment
CRL	Certificate revocation list
CSR	Certificate signing request
CSTA	Computer-supported telecommunications applications
CSV	Comma-separated values
DDDS	Dynamic delegation discovery system
DHCP	Dynamic Host Configuration Protocol
DMZ	Demilitarized zone
DN	Distinguished name
DNIS	Dialed number identification service
DNS	Domain name service
DOS	Denial of service
EIM	Enterprise instant messaging
ESD	Electrostatic discharge
ESGW	Emergency services gateway
ESQK	Emergency services query key
ESRN	Emergency services routing number

FQDN	Fully qualified domain name
GUI	Graphical user interface
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
I2	National Emergency Number Association defined VoIP solution
ICAP	Internet Calendar Access Protocol
ICMP	Internet Control Message Protocol
IM	Instant messaging
IP	Internet Protocol
JDBC	Java database connectivity
JMX	Java management extensions
JRE	Java runtime environment
LATA	Local access and transport area
LCS	Live Communications Server
LCR	Least-cost routing
LDAP	Lightweight Directory Access Protocol
LIS	Location information service
MAC	Media access control
MCS	Multimedia Communications Server
MIB	Management information base
MOS	Mean opinion score
MSAG	Master street address guide
MTU	Maximum transmission unit
NAPTR	Naming authority pointer
NAT	Network address translation
NENA	National Emergency Number Association
NIC	Network interface card
NS	Name server
NSE	Named signaling events
NTLM	NT Lan Manager
NTP	Network Time Protocol
OC	Office Communicator

OCI	Open Client Interface
ODBC	Open database connectivity
OTP	Over temperature protection
OVP	Over voltage protection
PBX	Private branch eXchange
PEM	Privacy-enhanced mail
PERL	Practical Extraction and Reporting Language
PING	Packet internet groper
PKCS#12	Public Key Cryptography Standard #12
PKI	Public Key Infrastructure
PSAP	Public safety answering point
PSCP	PuTTY secure copy
PSTN	Public switched telephone network
QOP	Quality of protection
QOS	Quality of service
RADIUS	Remote Authentication Dial-in User Service
RTC	Real-time collaboration
RTCP	Real-time Control Protocol
RTP	Real-time Transport Protocol
RTT	Round-trip time
SATA	Serial ATA
SCSI	Small computer system interface
SDK	Software development kit
SDP	Session Description Protocol
SFTP	Secure Shell File Transfer Protocol
SIMPLE	SIP Instant Messaging and Presence Leveraging Extension
SIP	Session Initiation Protocol
SIPS	Session Initiation Protocol over TLS
SLB	Server load balancing
SMB	Server message block
SNMP	Simple Network Management Protocol
SOA	Server of authority

SOAP	Simple Object Access Protocol
SQL	Structured Query Language
SRTP	Secure Real-time Transport Protocol
SRV	Server resource
SSH	Secure Shell
SSL	Secure socket layer
SSRC	Synchronization source
STUN	Simple Traversal of UDP over NATs
TCP	Transmission Control Protocol
TDM	Time division multiplexing
TGRP	Trunk group
TLS	Transport Layer Security
TOS	Type of service
TTL	Time to live
UPS	Uninterruptable power supply
US	User agent
UAC	User agent client
UAS	User agent server
UDP	User Datagram Protocol
UID	Unique identifier
URI	Uniform resource identifier
URL	Uniform resource locator
UTC	Universal coordinated time
VoIP	Voice over IP
VLAN	Virtual local area network
VPC	VoIP positioning center
VRRP	Virtual Router Redundancy Protocol
VSP	Virtual system partition
VXID	Virtual router interface ID
WAR	Web application resource
WAV	Waveform audio
WM	Windows Messenger

WSDL	Web Services Description Language
XML	Extensible Markup Language
XSL	Extensible Stylesheet Language

1. Managing and administering AA-SBC systems

About this chapter

The chapter describes the administrator tasks that you can perform when managing a new AA-SBC.

References

For detailed descriptions of the commands that you can use for administrative tasks, as well as instructions for using the management interfaces, refer to the *Net-Net OS-E – Objects and Properties Reference*.

For information on configuring policies, refer to the *Net-Net OS-E – Session Services Configuration Guide*.

Administrator and user roles

The *administrator* is any person who configures and manages the AA-SBC platform in the network.

The *user* is a SIP client, usually a VoIP call sender or receiver, of SIP messages that are transmitted to, and over the AA-SBC platform to a destination. A SIP user may have one or more SIP URIs in SIP sessions that traverse the platform between the user's originating SIP application or device and the SIP server endpoint (such as Microsoft LCS, IBM Sametime, Avaya, etc.). SIP clients who establish SIP sessions are subject to SIP policies that are configured by the AA-SBC administrator.

Enabling management access

When you create one or more administrative users, AA-SBC prompts for a username and password when anyone attempts to log in. Administrative users have read/write management access to the AA-SBC configuration file. Editing and saving the configuration file updates the AA-SBC configuration file named *cxc.cfg*. If desired, administrators can commit the configuration changes to the running AA-SBC configuration.

CLI session

The following CLI session creates a user and password (with permissions) for management access across the entire AA-SBC platform.

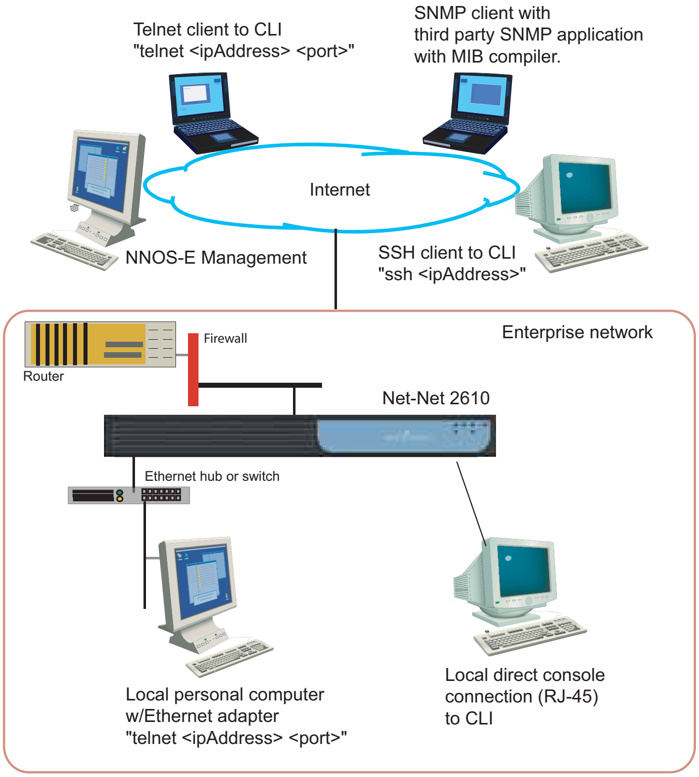
```
NNOS-E> config access
config access> config users
Creating ‘users’
config users> config user “jane doe”
Creating ‘user “jane doe”’
config user “jane doe”> set password abcXYZ
confirm:*******************
config user “jane doe”> set permissions access permissions grant
Creating ‘access\permissions grant’
config user “jane doe”> return
config users> return
config access> config permissions grant
Creating ‘permissions grant’
config permissions grant> set ftp enabled
config permissions grant> set cms enabled-web-only
config permissions grant> set cli normal
config permissions grant> set config enabled
config permissions grant> set call-logs enabled
config permissions grant> set actions enabled
config permissions grant> set status enabled
config permissions grant> set user-portal enabled
config permissions grant> set web-services enabled
```

If you are using the CMS to configure administrative users and permissions, use the CMS Access tab.

For more information on the **access** configuration object and the other properties that you can configure, refer to the *Net-Net OS-E – Objects and Properties Reference*.

Configuring management options

This section shows you how to set up the management options that allow you to configure the AA-SBC platform. The following image illustrates a sample network showing the supported management options.



Local console

If you are using a directly-attached local console or terminal to configure AA-SBC for the first time, use a terminal emulation program such as HyperTerminal to set the console parameters.

The following CLI session configures the console settings for communicating with the AA-SBC platform. The example session shows the console default settings.

CLI session

```
config> config box
config box> config console
config box> set rate 115200
config box> set data-bits 8
config box> set parity none
config box> set stop-bits 1
config box> set flow-control none
```

Telnet

Telnet is a standard TCP/IP-based terminal emulation protocol defined in RFC 854, *Telnet Protocol Specification.* Telnet allows a remote user to establish a terminal connection to the AA-SBC platform over an IP network. By default, the Telnet protocol is enabled at installation time. To allow connections over Telnet, you must configure those users who are allowed access to AA-SBC over Telnet.

The following CLI session configures the Telnet protocol on the local AA-SBC device, including the maximum number of concurrent Telnet sessions, the idle timeout period (in seconds) that ends a Telnet session due to inactivity, and the known TCP port for inbound and outbound Telnet messages.

CLI session

```
config box> config interface eth0
config interface eth0> config ip local
config ip local> config telnet
config telnet> set admin enabled
config telnet> set max-sessions 10
config telnet> set idle-timeout 600
config telnet> set port 23
```

Secure Shell (SSH)

Secure Shell (SSH) Server Version 2 on the AA-SBC platform 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). By default, SSH is enabled at installation time.

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 the AA-SBC platform, 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 parameters typically supported by SSH workstations. If you try to change a parameter that the AA-SBC does not support, you will receive a notification that the parameter setting failed.

CLI session

The following CLI session configures the SSH protocol on the local AA-SBC platform, including the maximum number of concurrent SSH sessions, the idle timeout period (in seconds) that ends an SSH session due to inactivity, and the known TCP port for inbound and outbound SSH messages.

```
config box> config interface eth0
config interface eth0> config ip local
config ip 1ocal> config ssh
config ssh> set admin enabled
config ssh> set max-sessions 10
config ssh> set idle-timeout 600
config ssh> set port 22
```

Web/HTTP

The AA-SBC Management System allows you to configure and manage the AA-SBC platform remotely using your web browser. .

The AA-SBC Management System interface supports all management capabilities provided by the CLI. Instead of entering information on a command line, you navigate menus and supply information in menu fields.

To manage the AA-SBC platform over the Web, enter the IP address of the management IP interface in the Internet Explorer **File/Open** command window and log in. For example:

http://192.168.124.1/

CLI session

The following CLI session enables Web access to the local AA-SBC platform and specifies the TCP port over which HTTPS traffic is sent and received on the IP interface.

```
config box> config interface eth0
config interface eth0> config ip local
config ip local> config web
config web> set admin enabled
config web> set protocol https 443
```

For detailed on using the CMS, refer to the *SIP Security and Management Solutions – System Management Reference*.

SNMP

The Simple Network Management Protocol (SNMP) allows you to communicate with the SNMP agent on the AA-SBC platform from a remote management station. SNMP allows you to retrieve information about managed objects on the platform as well as initiate actions using the standard and enterprise Management Information Base (MIB) files that Acme Packet makes available with the product software.

AA-SBC supports the SNMP versions SNMP v1 and SNMP v2c.

CLI session

The following CLI session enables SNMP access to the local AA-SBC platform, specifies the TCP port over which SNMP traffic is sent and received on the management interface, sets the SNMP community string, the SNMP version, and the target system IP address to which SNMP trap messages are forwarded.

```
config box> config interface eth0
config interface eth0> config ip local
config ip local> config snmp
config snmp> set admin enabled
config snmp> set port 161
config snmp> set version 2c
config snmp> set community private
config snmp> set trap-target 192.168.124.10
```

HTTP\SOAP\WSDL interface

The AA-SBC software includes a software development kit (SDK) to provide Web Services Description Language (WSDL) accessibility to AA-SBC.

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) is a communication protocol 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 an 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.

AA-SBC can perform two roles in the WSDL exchange:

- As a web service server, where an external client can make web service requests on the AA-SBC platform.
- As a web service client, where AA-SBC can make web service “call outs” to get location and policy information from an external service endpoint.

The WSDL document (and its imported schema files, such as cxc.xsd) define every possible request and response provided for the service, including error responses. Depending on how you choose to integrate with the AA-SBC platform, you can use the AA-SBC SDK (using Java) or you can simply take the WSDL document and generate tools in your desired language. Because web services are language independent, you can use virtually any modern language to generate the requests and the WSDL document defines what those requests need to look like for the receiving component.

For complete information on the WSDL interface, refer to the *Net-Net OS-E – Using the NNOS-E Management Tools*.

Working with the AA-SBC configuration file

All AA-SBC platforms use the startup configuration file named *cxc.cfg*. This file defines all aspects of the AA-SBC platform and its configuration in the network.

- Ethernet interfaces (and their IP addresses) connecting the platform to the Ethernet switches and the Internet
- Configured protocols, services, accounting and logging
- Policies that define the rules and conditions to match with SIP enterprise the carrier traffic requests.

Building the configuration file using the CLI

The AA-SBC configuration file (*cxc.cfg*) is made up of configuration objects and property settings that control how the system processes and manages SIP traffic. As you open these objects and set properties using the CLI (or the CMS), AA-SBC builds a configuration hierarchy of objects that are applied to SIP sessions. You can display this configuration hierarchy using the **show** and **show -v** commands.

For new users, as well as for users who are adding functionality to their configuration, you will need to open configuration objects using the **config** command to enable the default settings for those objects, even if you choose not to edit any of their associated properties. For example, if you need to enable the **ICMP** protocol and its default settings, you simply open the object and execute **return**, as shown in the session below. Notice that the ICMP object has been added to the configuration hierarchy at the end of the session on the eth4 interface.

CLI session

```
config> config box interface eth4
config interface eth4> config ip 172.26.2.14
config ip 172.26.2.14> config icmp
config ip 172.26.2.14> return
config interface eth4> return
config box> return
config> show -v
   interface eth4
     admin enabled
     mtu 1500
     arp enabled
     speed 1Gb
     duplex full
     autoneg enabled
     ip 172.26.2.14
      admin enabled
      ip-address dhcp
      geolocation 0
      metric 1
      classification-tag
      security-domain
      address-scope
      filter-intf disabled
      icmp
       admin enabled
       limit 10 5
```

Removing objects from the configuration file using the CLI

To remove an object from the configuration hierarchy, use the CLI or CMS **delete** command. For example, the CLI session below deletes the IP interface 172.26.1.14 from the configuration hierarchy:

CLI session

```
config> config box interface eth4
config interface eth4> delete ip 172.26.1.14
```

Editing and saving the configuration file using the CLI

There are three levels of configuration—the working config which keeps a record of configuration edits, the running configuration which is used by the system, and the startup configuration file from which the system boots.

1. The startup, or default, config is saved to the /cxc/cxc.cfg file. When the AA-SBC platform starts, it loads the startup config into the running config. Use the `save` command, either at the config prompt (`config>`) or at the top–level prompt `Net-Net>)` by default), to save the running config to the startup config.

2. The running config is the current operational configuration. You can display the running config using the following command:

   ```
   Net-Net> config show -v
   ```

   Edit the running config using the CLI `config` command, or NNOS-E Management application. You can save the running config to a file (either the startup config file or a different file) using the `config save` command.

3. When you edit a configuration object, you get a working copy of that object. The working config maintains a record of all configuration changes you have made since the last save to the running config. However, your changes are not applied to the running config until you explicitly commit them. While you're editing an object, the show command displays your working copy. Use the `commit` command, or **exit** from config mode and answer **yes** to the prompt, to save changes from the working configuration to the running configuration.

For detailed information on using the CLI and other management services that allow you to edit the config file, refer to the *Net-Net OS-E – Objects and Properties Reference*.

Creating SIP users and passwords

The **user** configuration object allows you to define the users who can pass SIP traffic on this virtual system partition (VSP). (Refer to refer to "AA-SBC virtual system partitions" for more information about AA-SBC VSPs). The users object only applies if your SIP configuration requires local authentication in the **default-session-configuration** object under VSP, or in the **session-configuration** object under the policy configuration object.

When you enable the local authentication file, you configure AA-SBC to prompt those users that are passing SIP traffic to log in. The user name and password tag 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.

CLI session

The following CLI session creates a locally authenticated SIP user.

```
NNOS-E> config vsp
config vsp> config user bob-pc@companySierra.com
Creating ‘user bob-pc@companySierra.com’
config user bob-pc@companySierra.com> set admin enabled
config user bob-pc@companySierra.com> set password-tag abcXYZ
```

Unlike AA-SBC administrative users, SIP users who log in with a valid user name and password do not have read/write access to the AA-SBC configuration file.

Customizing the CLI

AA-SBC software allows you to customize the CLI to accommodate the type of display you are using, as well as change the default top-level prompt that is pre-configured with the platform.

CLI session

The following CLI session sets the number of rows that the CLI displays in a single page to 24 lines, and resets the default top-level prompt from Net-Net> to *boston1>*, and sets an optional text banner to appear when you start the CLI.

```
config box> config cli
config cli> set display paged 24
config cli> set prompt boston1>
config cli> set banner text
```

To temporarily change the CLI display mode with changing the default configuration, use the **display** command at the top level of the CLI.

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

Whenever you use paged output, the --More-- prompt accepts 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 change from paged output to continuous scrolled output, enter the following command:

```
config cli> set display scrolled
```

Setting AA-SBC global properties

You can configure global text properties associated with each AA-SBC platform in the network. These global text properties include:

- hostname
- name
- description
- contact
- location
- timezone

CLI session

The following CLI session enables the AA-SBC administrative state, and sets the optional text descriptions associated with this AA-SBC platform.

```
NNOS-E> config box
config box> set admin enabled
config box> set hostname company.boston1.companySierra.com
config box> set name boston1
config box> set description Net-NetMasterBoston
config box> set contact adminFred
config box> set location corpDataCenter
config box> set timezone Pacific
```

AA-SBC virtual system partitions

AA-SBC's virtual system partition (VSP) is the part of the system that holds the comprehensive customer-defined configuration that controls how the system processes, stores, directs, and routes SIP traffic. The VSP is where you can create session configurations, registration and dial plans, and policies that handle SIP REGISTER and SIP INVITE traffic (and other SIP methods) that AA-SBC will receive and forward to a SIP call destination, authentication and accounting database, VoIP service provider or carrier, enterprise server, and so on.

The VSP configuration uses objects and properties that control the majority of the AA-SBC platform functionality.

Specifying management preferences

The **cms-preferences** object allows you to configure enumeration text strings to network, database, and SIP objects that support extensions, as well as preferences for reverse DNS, trap polling intervals, phone path mapping, and the cluster and box summary information to include on the Status summary page.

CLI session

The following CLI session configures the `securityDomain` and the `sipHeaderNameEnum` strings, how frequently (in seconds) to check for SNMP traps

```
NNOS-E> config preferences
config preferences> config cms-preferences
config cms-preferences> set enum-strings securityDomain untrusted
config cms-preferences> set enum-strings sipHeaderNameEnum
   accept-encoding
config cms preferences> set trap-poll-interval 60
```

For more information on configuring the optional enumeration strings, refer to *Net-Net OS-E – Objects and Properties Reference*.

Specifying DOS query preferences

Denial of service (DOS) attacks are designed to disable networks by flooding them with useless traffic. Net-Net OS-E platform 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.

CLI session

The following CLI session opens the **dos-queries** object and a named sip-query (companySierra), followed by the sip-query options that control how the query displays and sorts DOS traffic:

```
NNOS-E> config preferences
config preferences> config dos-queries
config dos-queries> config sip-query companySierra
Creating ‘sip-query companySierra’
config sip-query companySierra> set description “SIP-layer queries”
config sip-query companySierra> set admin enabled
config sip-query companySierra> set select content-type
config sip-query companySierra> set group session-id
config sip-query companySierra> set sort timestamp
config sip-query companySierra> set order ascending
```

For more information on configuring the DOS query preferences, refer to the *Net-Net OS-E – Session Services Configuration Guide* and the *Net-Net OS-E – Objects and Properties Reference*.

Restarting and shutting down the system

At times, you may need to shut down or restart the system. The AA-SBC should only be shut down or restarted via the System Platform web console.

- To shut down the system completely, press the On/Off button on the chassis to OFF.
- To perform a warm or cold restart or a system halt, use the **restart** command. A **restart warm** resets the AA-SBC application software; a **restart cold** reboots the platform, **restart halt** suspends AA-SBC operation without rebooting or restarting.
- To simultaneously warm restart all systems in the network cluster, use the **restart cluster** command.

Caution: Always save your configuration before you shut down or restart the system.

When you restart the AA-SBC device, the system uses the latest saved configuration file. If you do not save a configuration prior to a reboot or shutdown, you lose any changes you made since you last saved the configuration file.

CLI session

The following session performs a Net-Net OS-E platform warm restart:

```
NNOS-E> restart warm
```

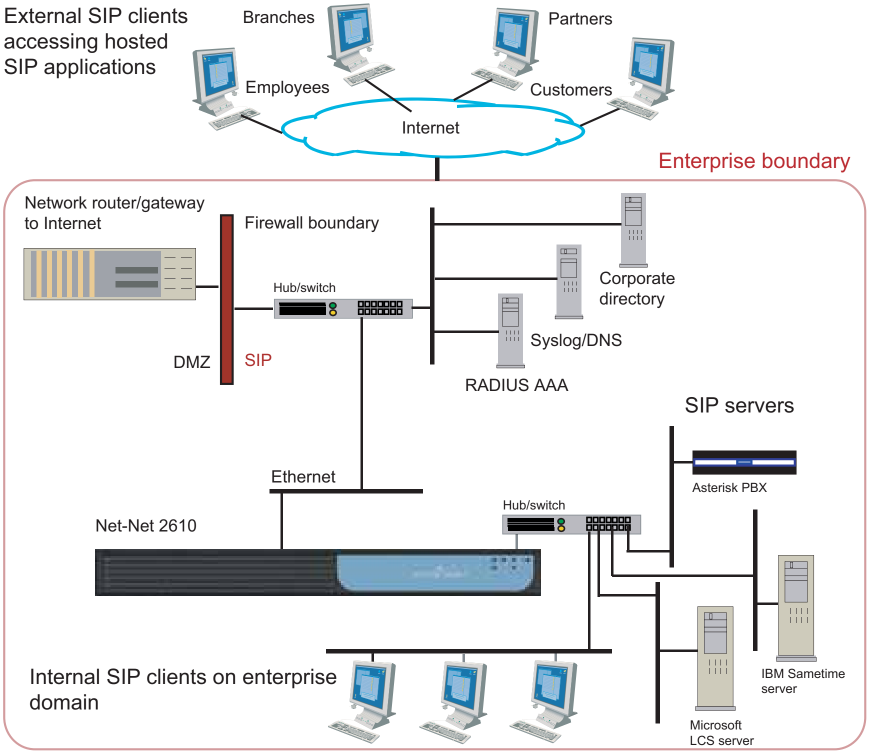
2. Enabling AA-SBC interfaces and protocols

About this chapter

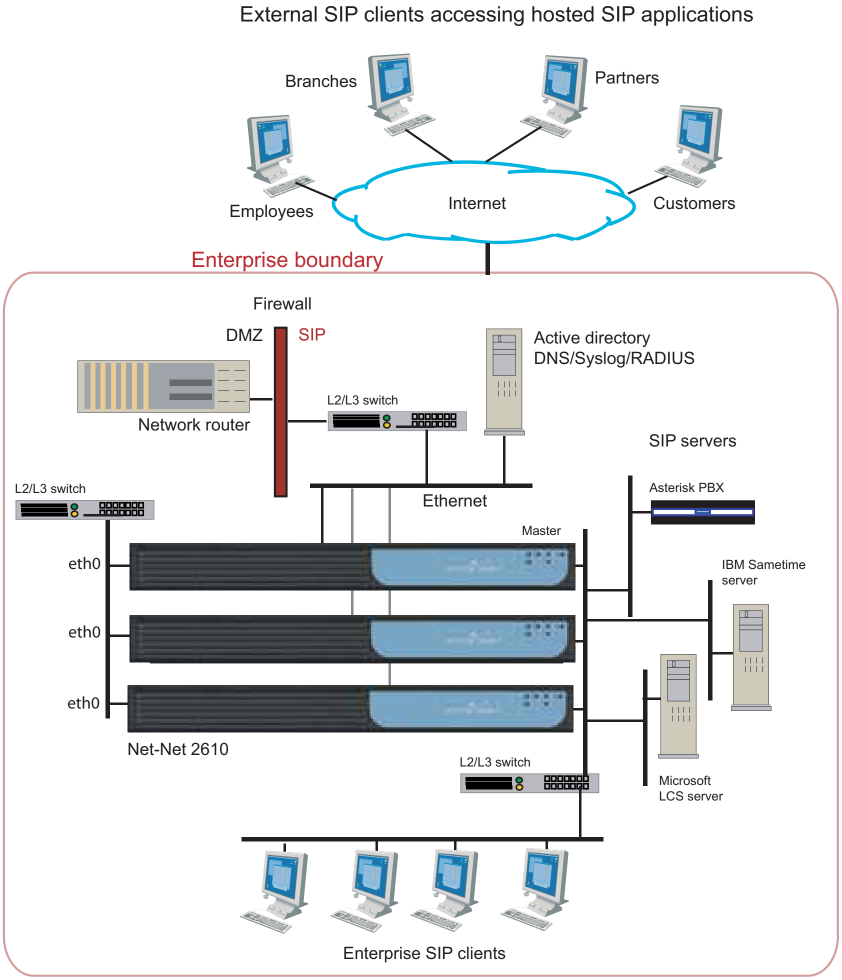
The chapter describes network interfaces and the protocols that you can enable on AA-SBC platforms.

AA-SBC sample networks

The following image illustrates a sample enterprise network with a single AA-SBC device.



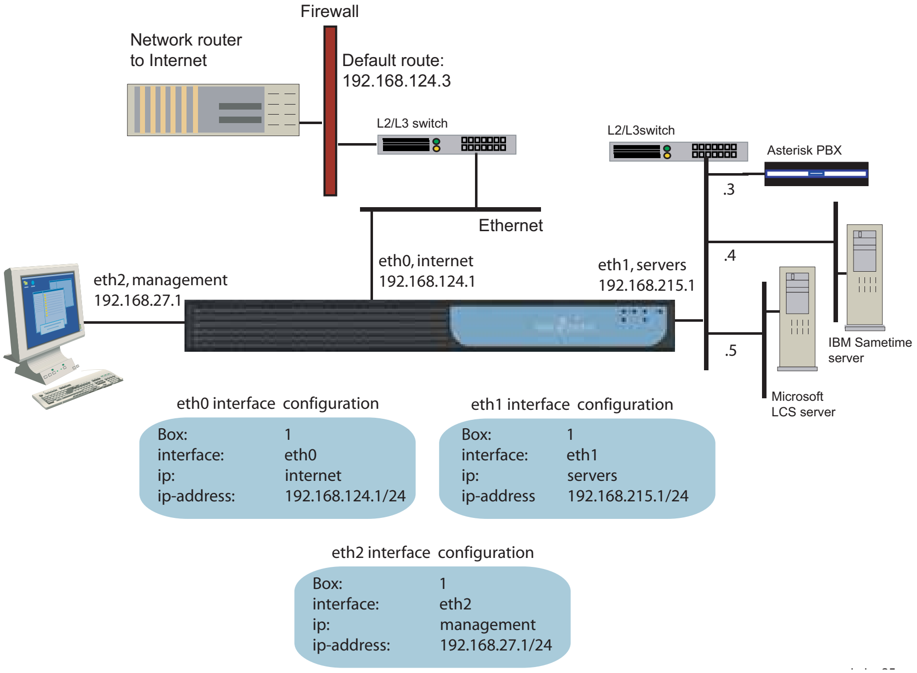
The following image illustrates a sample enterprise that uses a AA-SBC cluster.



Configuring AA-SBC IP interfaces

AA-SBC physical interfaces include multiple Ethernet 1000 Mbps auto-negotiation interfaces, such as eth0, eth1, eth2, and eth3. The number of interfaces depends on the specific platform you are using.

AA-SBC software uses IP objects — which are assigned a name by the system administrator — to uniquely identify IP connections. Each physical Ethernet interface can contain up to 255 uniquely named IP objects. The following image illustrates a sample network with one named IP object on each physical Ethernet interface.



eth0 interface configuration

```
Box:             1
interface:       eth0
ip:              internet
ip-address:      192.168.124.1/24
```

eth1 interface configuration

```
Box:             1
interface:       eth1
ip:              servers
ip-address       192.168.215.1/24
```

eth2 interface configuration

```
Box:             1
interface:       eth2
ip:              management
ip-address:      192.168.27.1/24
```

CLI session for eth0

The network on physical interface eth0 uses the IP object that the system administrator named *internet*. The *internet* object specifies the IP address that connects to the external Internet local gateway using a default route.

```
NNOS-E> config cluster
config cluster> config box 1
config box 1> config interface eth0
config interface eth0> config ip internet
Creating ‘ip internet’
```

```
config ip internet> set ip-address static 192.168.124.1/24
config ip internet> return

config interface eth0> config ip internet
config ip internet> set ip-address static 192.168.124.2/24
config ip internet> config routing
config routing> config route internetGateway
config route internetGateway> set destination default
config route internetGateway> set gateway 192.168.124.3
```

CLI session for eth1

The network on physical interface eth1 uses the IP object named *servers*. The static IP address points to the SIP destination servers on the same network subnet, connected over Ethernet switch.

```
NNOS-E> config cluster
config cluster> config box 1
config box 1> config interface eth1
config interface eth1> config ip servers
config ip servers> set ip-address static 192.168.215.1/24
config ip servers> return
config interface eth1>
```

CLI session for eth2

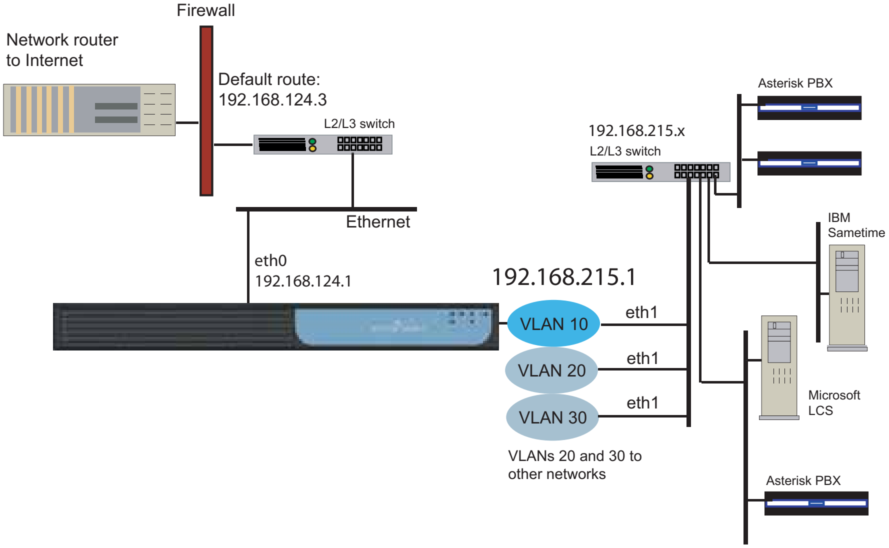
The network on physical interface eth2 uses the defined IP object named *management*. The management object specifies the IP address over which management traffic is carried, such as remote CLI session over Telnet, or a AA-SBC Management System session.

```
NNOS-E> config cluster
config cluster> config box 1
config box 1> config interface eth2
config interface eth2> config ip management
config ip internet> set ip-address static 192.168.27.1/24
```

Creating VLANs

AA-SBC virtual LANs (VLANs) provide Layer 2 partitions to the communications servers. Creating one or more VLANs allows you to group LAN segments so that they appear to be on the same Layer 2 network. Each VLAN is identified by a VLAN ID, and ID must be unique within the physical AA-SBC device. This means that multiple logical AA-SBC devices (called VSPs) cannot use the same VLAN IDs. VLAN IDs can be in the range 1 to 4096.

The following image illustrates a sample VLAN configuration.



CLI session

The following CLI session configures the VLAN 10 network. VLAN 10 supports three separate physical IP networks, and all appearing as if they are on the same Layer2 network.

```
NNOS-E> config cluster
config cluster> config box 1
config box 1> config interface eth1
config interface eth1> config vlan 10
Creating ‘vlan10’
config vlan 10> config ip servers
Creating ‘ip servers’
config ip servers> set ip-address static 192.168.215.1/24
config ip servers> return
```

Applying routing and classification tags

The system uses classification tags to classify incoming traffic and routing tags to control the egress route for a specific service type. Tags allow the IP routing table in Session Manger to be segmented into multiple routing tables. Once an interface has a configured routing tag, the interface is removed from the “null” (or system routing table).

When traffic comes arrives at a AA-SBC platform on an identified interface, you can direct that traffic to a specific egress interface to the destination. This means that you would configure a **classification-tag** on the incoming interface that matches the **routing-tag** on the desired egress interface.

You can create multiple routing tags on the same named IP interface. However, only one classification tag is allowed per IP interface. Both routing and classification tags are case sensitive with the following configuration properties:

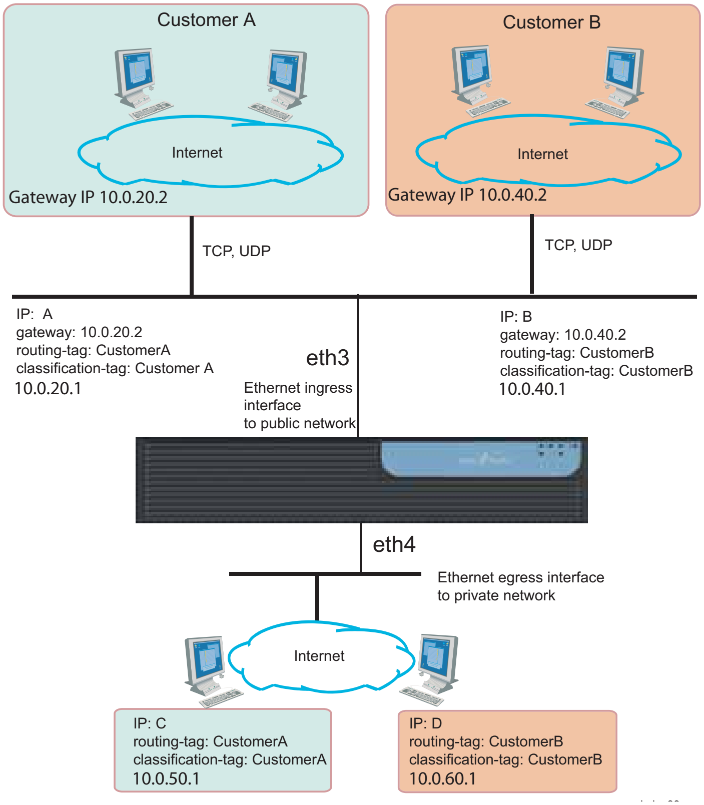
- **routing-tag** — Associates all the routes configured on an interface with this **routing-tag** and creates a service route table based on the routing-tag for each service enabled on this interface. The **routing-tag** applies to the *egress* interface over which AA-SBC forwards service traffic. Once 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).

  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

- **classification-tag** — Creates a tag associated with inbound traffic on this interface. This means that you must configure a **classification-tag** on the *ingress* interface over which the AA-SBC domain initially receives the traffic, matching the **routing-tag**. (Classification tags in the session configuration **routing-settings** object also must match this routing tag set in the **ip** object.

Note: 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.

The following image illustrates a sample network where routing and classification tags are configured on the ingress and egress AA-SBC interfaces, followed by sample configuration sessions for ingress and egress IP instances.



CLI sessions for “IP A” and “IP B” ingress networks on eth3

The following CLI sessions create the ingress side of the network illustrated in the image above, including the IP addresses, routing and classification tags, SIP settings, and a route to the IP using the gateways at IP addresses at 10.0.20.2 and 10.0.40.2. AA-SBC uses classification tags to classify incoming traffic and routing tags to control the egress route. Configure a **classification-tag** on the incoming interface that matches the **routing-tag** on the egress interface.

```
NNOS-E> config cluster
config cluster> config box 1
config box 1> config interface eth3
Creating ‘interface eth3’
config interface eth3> config ip A
```

```
Creating ‘ip A’
config ip A> set ip-address static 10.0.20.1/24
config ip A> set classification-tag CustomerA
config ip A> set routing-tag CustomerA
config ip A> config sip
config sip> set admin enabled
config sip> set nat-translation enabled
config sip> set udp-port 5060
config sip> set tcp-port 5060
config sip> return
config ip A> config icmp
config icmp> return
config ip A> config routing
config routing> config route default
Creating ‘route default’
config route default> set gateway 10.0.20.2
config route default> return
config routing> return
config ip A> return
```

```
NNOS-E> config cluster
config cluster> config box 1
config box 1> config interface eth3
Creating ‘interface eth3’
config interface eth3> config ip B
Creating ‘ip B’
config ip B> set ip-address static 10.0.40.1/24
config ip B> set classification-tag CustomerB
config ip B> set routing-tag CustomerB
config ip B> config sip
config sip> set admin enabled
config sip> set nat-translation enabled
config sip> set udp-port 5060
config sip> set tcp-port 5060
config sip> return
config ip B> config icmp
config icmp> return
config ip B> config routing
config routing> config route default
Creating ‘route default’
config route default> set gateway 10.0.40.2
config route default> return
config routing> return
config ip B> return
```

CLI sessions for “IP C” and “IP D” egress networks on eth4

The following CLI sessions create the egress side of the network illustrated in the image above, including the IP addresses, routing and classification tags, SIP settings, and a default route. AA-SBC uses classification tags to classify incoming traffic and routing tags to control the egress route. Configure a **classification-tag** on the incoming interface that matches the **routing-tag** on the egress interface.

```
NNOS-E> config cluster
config cluster> config box 1
config box 1> config interface eth4
Creating ‘interface eth4’
config interface eth4> config ip C
Creating ‘ip C’
config ip C> set ip-address static 10.0.50.1/24
config ip C> set classification-tag CustomerA
config ip C> set routing-tag CustomerA
config ip C> config sip
config sip> set admin enabled
config sip> set nat-translation enabled
config sip> set udp-port 5060
config sip> set tcp-port 5060
config sip> return
config ip C> config icmp
config icmp> return
config ip C> config routing
config routing> config route default
Creating ‘route default’
config route default> set destination default
config route default> return
```

```
NNOS-E> config cluster
config cluster> config box 1
config box 1> config interface eth4
Creating ‘interface eth4’
config interface eth4> config ip D
Creating ‘ip D’
config ip D> set ip-address static 10.0.60.1/24
config ip D> set classification-tag CustomerB
config ip D> set routing-tag CustomerB
config ip D> config sip
config sip> set admin enabled
config sip> set nat-translation enabled
config sip> set udp-port 5060
config sip> set tcp-port 5060
config sip> return
config ip D> config icmp
config icmp> return
config ip D> config routing
config routing> config route default
Creating ‘route default’
config route default> set destination default
config route default> return
```

Notes on routing and classification tags

- Separate routing tables are maintained for the SIP and media service:
  - — IP interfaces without SIP ports enabled will not appear in the SIP table
  - — IP interfaces without media ports enabled will not appear in the media table.
- SIP or media traffic that is classified by a tag will only use the routing information and interfaces that have been configured with that routing tag.
- An address of record (AOR) will be assigned an ingress tag IF the REGISTER for that AOR
  - — Ingresses on an IP interface with a configured **classification-tag**.
  - — Matches a policy or registration-plan that applies a session configuration that has the **ingress-classification-tag** property configured. This overwrites the IP interface **classification-tag**, if configured.
  - — Matches a calling-group. The **classification-tag** for the calling-group is only applied if a tag has not been assigned using the IP or session configuration.
- Traffic can be assigned an egress tag as follows:
  - — From an ingress tag.
  - — From a matching policy or dial-plan that applies a session configuration that has the **egress-classification-tag** configured. This overwrites the **classification-tag** configured on the interface.
  - — From a server or carrier with the routing-tag configured, overwriting all other tags.

Related commands

To assist troubleshooting, use the following commands from the AA-SBC prompt to display information about tag-routing.

- **show services-routing —** Displays routing tables for all tags.
- **show services-routing-tables —** Displays all configured tags.
- **service-route-lookup —** To view the destination where AA-SBC routed a call**.**

Configuring overlapping IP networks and tag routing

A preferred method for creating networks, with overlapping IPs is to configure VLANs with routing tags. A routing tag associates all the routes configured on an interface and creates a service route table based on the tag for each service enabled the interface. Routing tags apply to the egress interface over which AA-SBC forwards service traffic.

To perform tag routing, do the following:

1. Configure a **classification-tag** on the ingress interface over which AA-SBC initially receives service traffic. The classification tag must match the configured **routing-tag**; each IP interface can have multiple routing tags.
2. Set the **egress-classification-tag** property under the **session-config/ routing-settings** when sending service traffic to servers and carriers.

CLI session for Ethernet public and private sides of network

The following CLI session configures the AA-SBC *public* IP Ethernet interface and SIP settings.

```
NNOS-E> config cluster
config cluster> config box 1
config box 1> config interface eth3
Creating ‘interface eth3’
config interface eth3> config ip public
Creating ‘ip public’
config ip public> set ip-address static 10.0.10.1/24
config ip public> config sip
config sip> set admin enabled
config sip> set nat-translation enabled
config sip> return
```

The following CLI session configures the AA-SBC *private* IP Ethernet interface and SIP settings.

```
NNOS-E> config cluster
config cluster> config box 1
config box 1> config interface eth4
Creating ‘interface eth4’
config interface eth4> config ip private
Creating ‘ip private’
config ip private> set ip-address static 10.0.20.1/24
config ip private> config sip
config sip> set admin enabled
config sip> set nat-translation enabled
config sip> return
```

CLI sessions for Customer-A and Customer-B networks

The following CLI sessions create the VLANs to the Customer-A and Customer-B networks, including the IP addresses, routing and classification tags, SIP settings, and a route to the IP using the gateways at IP addresses at 10.0.1.50 and 10.0.1.60. AA-SBC uses classification tags to classify incoming traffic and routing tags to control the egress route. Configure a **classification-tag** on the incoming interface that matches the **routing-tag** on the egress interface.

```
config interface eth3> config vlan 10
Creating ‘vlan 10’
config vlan 10> config ip 10.0.1.1
Creating ‘10.0.1.1’
config ip 10.0.1.l> set ip-address static 10.0.1.1/24
config ip 10.0.1.l> set classification-tag vlan10
config ip 10.0.1.1> set routing-tag vlan10
config ip 10.0.1.1> config sip
config sip> set nat-translation enabled
config sip> set udp-port 5060
config sip> set tcp-port 5060
config sip> return
config ip 10.0.1.1> config icmp
config icmp> return
config ip 10.0.1.1> config routing
config routing> config route default
Creating ‘route default’
config route default> set gateway 10.0.1.50
config route default> return
config routing> return
config ip 10.0.1.1> return

config interface eth3> config vlan 20
Creating ‘vlan 20’
config vlan 20> config ip 10.0.1.1
Creating ‘10.0.1.1’
config ip 10.0.1.l> set ip-address static 10.0.1.1/24
config ip 10.0.1.l> set classification-tag vlan20
config ip 10.0.1.1> set routing-tag vlan20
config ip 10.0.1.1> config sip
config sip> set nat-translation enabled
config sip> set udp-port 5060
config sip> set tcp-port 5060
config sip> return
config ip 10.0.1.1> config icmp
config icmp> return
config ip 10.0.1.1> config routing
config routing> config route default
Creating ‘route default’
config route default> set gateway 10.0.1.60
config route default> return
config routing> return
config ip 10.0.1.1> return
```

CLI session for the internal private network

The following CLI session creates the VLAN to the internal *private* network, including the private IP address, routing and classification tags, SIP settings, and a default route to the public IP interface at 10.0.20.1. AA-SBC uses classification tags to classify incoming traffic and routing tags to control the egress route. Configure a **classification-tag** on the incoming interface that matches the **routing-tag** on the egress interface.

```
config interface eth4> config vlan 30
Creating ‘vlan 30’
config vlan 10> config ip 10.0.20.1
Creating ‘10.0.20.1’
config ip 10.0.20.l> set ip-address static 10.0.20.1/24
config ip 10.0.20.l> set classification-tag MAIN
config ip 10.0.20.1> set routing-tag MAIN
config ip 10.0.20.1> config sip
config sip> set nat-translation enabled
config sip> set udp-port 5060
config sip> set tcp-port 5060
config sip> return
config ip 10.0.20.1> config icmp
config icmp> return
config ip 10.0.20.1> config routing
config routing> config route default
Creating ‘route default’
config route default> set destination default
config route default> return
config routing> return
config ip 10.0.1.1> return
```

CLI session for the session-config-pool

The following CLI session creates two session configuration entries for handling egress traffic from Customer-A and Customer-B to AA-SBC. The session-config-pool is for any traffic routed to the private network. The **egress-classification-tag** property, which needs to match the appropriate VLAN routing-tag on VLAN 30, selects the interface to the private network.

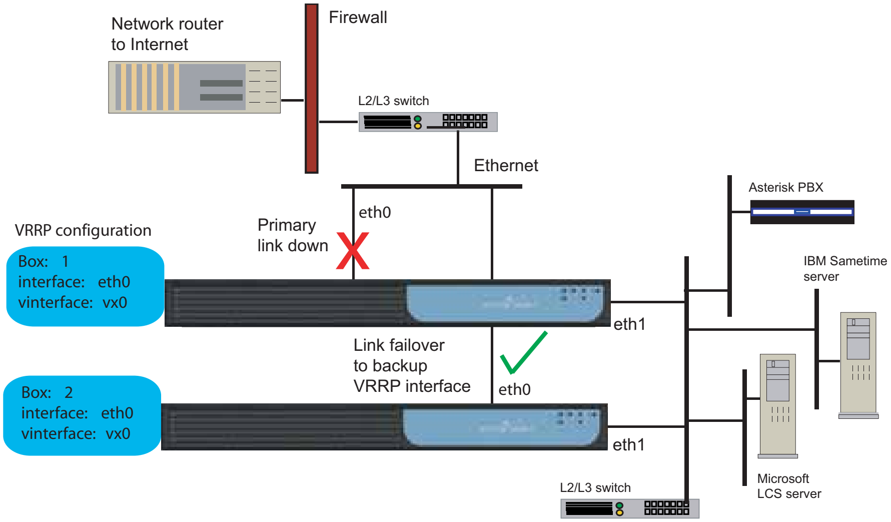
```
config> config vsp session-config-pool
config session-config-pool> config entry "Customer-A"
Creating entry “Customer A”
config entry “Custom A”> config routing-settings
config routing-settings> set egress-classification-tag MAIN
config routing-settings> return
config entry “Custom A”> return
config session-config-pool> config entry "Customer-B"
Creating entry “Customer B”
config entry “Custom B”> config routing-settings
config routing-settings> set egress-classification-tag MAIN
```

Configuring VRRP

The Virtual Router Redundancy Protocol (VRRP) provides redundancy of IP interfaces within a AA-SBC cluster. The configuration for IP interfaces includes a list of box/interface pairs. The first pair in this list is the *primary interface*. The second pair in the list is the *backup interface* and will take over if the primary goes down. You can configure additional levels of redundancy by specifying more box/interface pairs of lower priority. Priority is based on the positioning of the **set host-interface** command.

VRRP also provides redundancy of master services within a cluster. Each master service, including directory, database, and accounting, can be configured with a list of locations (box numbers within the cluster). The first location, such as box 1, is the primary; the second location (box 2) takes over if the primary fails. Specifying more locations in the list creates additional levels of redundancy.

The following image illustrates a sample network where VRRP reroutes traffic around a failed interface.



If the master VRRP interface becomes unavailable, the VRRP election protocol enables a backup VRRP interface to assume mastership using the next prioritized interface in the list. However, if the original master VRRP interface (the interface with the highest priority) should once again become available, VRRP returns mastership to that interface.

See RFC 2338, *Virtual Router Redundancy Protocol*, for detailed information about this protocol.

CLI session

The following CLI session creates two VRRP virtual interfaces (vx0 and vx1), and configures the physical host interfaces associated with each vinterface. On the vx0 interface, physical interface eth0 on box 1 will failover to eth0 on box 2, and then to eth0 on box 3. Note that each VRRP interface has its own IP (or VLAN) configuration.

CLI session

```
NNOS-E> config cluster
config cluster> config vrrp
config vrrp> config vinterface vx0
config vinterface vx0> set host-interface cluster box 1 interface eth0
config vinterface vx0> set host-interface cluster box 2 interface eth0
config vinterface vx0> set host-interface cluster box 3 interface eth0
config vinterface vx0> config ip name
Creating ‘ip name’
config ip name> set ip-address static 1.1.1.1/24
config ip name> return
config vinterface vx0> return

config vrrp> config vinterface vx1
config vinterface vx1> set host-interface cluster box 3 interface eth1
config vinterface vx1> set host-interface cluster box 4 interface eth1
config vinterface vx1> config ip name
Creating ‘ip name’
config ip name> set ip-address static 1.1.1.2/24
config ip name> return
config vinterface vx0> return
```

See RFC 2338, *Virtual Router Redundancy Protocol*, for detailed information about VRRP.

When configuring VRRP backing interfaces, you should only have no more than two different AA-SBCs on the host list. You can, however, have more than one interface configured per box without any problems.

Here are some examples to illustrate acceptable and not acceptable configurations.

Not acceptable: There are interfaces from three different AA-SBCs listed for this VX interface. You should only have two AA-SBCs backing a VX.

```
config vrrp
 config vinterface vx10
  set group 1
  set host-interface cluster\box 1\interface eth1
  set host-interface cluster\box 2\interface eth1
  set host-interface cluster\box 3\interface eth1
  config ip 10.1.1.1
  return
 return
return
```

Not acceptable: There are interfaces from three different AA-SBCs listed for this VX interface and **preempt=true** is configured. This configuration is not supported at this time and will result in inconsistent behavior for the VX interface.

```
config vrrp
 config vinterface vx10
  set group 1
  set preempt true
  set host-interface cluster\box 1\interface eth1
  set host-interface cluster\box 2\interface eth1
  set host-interface cluster\box 3\interface eth1
  config ip 10.1.1.1
  return
 return
return
```

Acceptable: There are only two AA-SBCs listed as hosts for this VX.

```
config vrrp
 config vinterface vx10
  set group 1
  set host-interface cluster\box 1\interface eth1
  set host-interface cluster\box 2\interface eth1
  config ip 10.1.1.1
  return
 return
return
```

Acceptable: There are only two AA-SBCs listed as hosts for this VX, but each AA-SBC has two host interfaces configured on it.

```
config vrrp
 config vinterface vx10
  set group 1
  set host-interface cluster\box 1\interface eth1
  set host-interface cluster\box 1\interface eth2
  set host-interface cluster\box 2\interface eth1
  set host-interface cluster\box 2\interface eth2
  config ip 10.1.1.1
  return
 return
return
```

In either of these last two acceptable examples, it is okay to configure **preempt=true**.

Configuring signaling failover

AA-SBC platforms use signaling failover to preserve signaling sessions in a high-availability cluster. The cluster **master-service** maintains the signaling state of connections cluster-wide. With signaling failover, the signaling state information is transferred to the AA-SBC device taking over the signaling stream.

Note: The call must be connected (at the SIP level) in order for signaling failover to take place. Signaling states prior to the “connected” state are not maintained in the cluster wide state table. Additionally, for TCP and TLS connections, the user agent must re-establish the connection once the failover has occurred. Since TCP/TLS are connection-oriented protocols, signaling state information is not maintained across failover. If TLS is used, the appropriate certificate must be loaded on the AA-SBC platforms in the cluster.

Signaling information is maintained so that accurate call logs are recorded at the end of the call.

Note: If there is a failure at the AA-SBC platform holding the call log database, information will be lost.

Use the AA-SBC **show signaling-sessions** command to display failover state information.

CLI session

```
NNOS-E> config cluster
config cluster> set share-signaling-entries true
```

The **share-signaling-entries** property specifies whether or not all AA-SBC devices in a cluster exchange active SIP session information. When set to *true*, the AA-SBC devices exchange data. If the primary link then goes down, a backup link can use SIP session information from the primary device to handle existing calls.

The **share-signaling-entries** property should be set to *true* if you have configured VRRP (to provide the redundancy support). If you have VRRP enabled and configured, and if **share-signaling-entries** is set to true, signaling failover can take place.

Configuring Web interface settings

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.

CLI session

```
NNOS-E> config cluster
config cluster> config box 1
config box 1> config interface eth0
config interface eth0> config ip boston1
config ip boston1> config web
config web> set admin enabled
config web> set protocol https 443 0 “vsp tls certificate cxc.cert.com”
config web> set trap-target enabled
```

Configuring Web services

The **web-service** 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) 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 the AA-SBC device using HTTPS.

CLI session

```
NNOS-E> config cluster
config cluster> config box 1
config box 1> config interface eth0
config interface eth0> config ip boston1
config ip boston1> config web-service
config web-service> set admin enabled
config web-service> set protocol https 443 0 “vsp tls certificate
 cxc.company.com”
```

For detailed information on WSDL, refer to the *Net-Net OS-E – Using the NNOS-E Management Tools*.

Enabling ICMP and setting rate limits

The Internet Control Message Protocol (ICMP), defined in RFC 792, is a TCP/IP protocol that determines whether a destination is unreachable. Using error and control messages between an host and an Internet gateway, ICMP verifies the validity of an IP address.

You can limit the rate at which ICMP messages are received on the AA-SBC device by setting ICMP rate and burst limits that prevent flooding of ICMP messages on the network. The rate setting is the maximum number of ICMP destination unreachable messages that the device can receive per second; the burst setting is the rate by which the number of ICMP messages that are discarded per second. Configuring the burst setting to a number lower than the rate setting will prevent ICMP message flooding.

CLI session

The following CLI session enables ICMP on the specified interface and sets ICMP rate and burst limits.

```
NNOS-E> config cluster box 1
config box 1> config interface eth0
config interface eth0> config ip boston1
Creating ‘ip boston1’
config ip boston1> config icmp
config icmp> set admin enabled
config icmp> set limit 12 6
```

Enabling NTP and BOOTP servers

By default, Network Time Protocol (NTP) and BOOTP services are enabled. The AA-SBC device uses NTP to synchronize time with external and local clocks using an NTP server, and the BOOTP protocol to allow a AA-SBC network client to learn its own IP address and boot information from a BOOTP server.

If addition to configuring NTP and BOOTP clients, you need to ensure that the NTP and BOOTP services are enabled on AA-SBC IP interfaces.

CLI session

The following session enables BOOTP services on the specified AA-SBC IP interface and port number.

```
NNOS-E> config cluster box 1
config box 1> config interface eth0
config interface eth0> config ip boston1
Creating ‘ip boston1’
config ip boston1> config bootp-server
config bootp-server> set admin enabled
config bootp-server> set port 67
```

The following session enables NTP services on the specified AA-SBC IP interface.

```
NNOS-E> config cluster box 1
config box 1> config interface eth0
config interface eth0> config ip boston1
Creating ‘ip boston1’
config ip boston1> config ntp-server
config ntp-server> set admin enabled
```

Configuring the Bootstrap Protocol (BOOTP) clients

The BOOTP commands allow you to configure the Bootstrap Protocol (BOOTP) client and server settings in a AA-SBC network cluster. BOOTP, 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 network cluster, a BOOTP client requests its own IP address from the AA-SBC BOOTP server, 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 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.

CLI session

The following session configures a bootp client on the AA-SBC device. The session enables the bootp client, and sets the known bootp client and server ports for bootp requests and responses. UDP port 68 is the known bootp client port; UDP port 67 is the known bootp server port.

```
config box> config bootp-client
config bootp-client> set admin enabled
config bootp-client> set client-port eth1 68
config bootp-client> set server-port eth0 67
```

Configuring SIP

For SIP applications running over networks, you need to enable the Session Initiation Protocol (SIP) on the AA-SBC IP interfaces. By default, the SIP protocol is enabled. However, you do need to configure the SIP operation mode, set the UDP, TCP, and TLS ports to use when listening for SIP messages, and include any certificates (generated and imported from a certificate authority) to be associated with the SIP interface.

- In *proxy* mode, the AA-SBC device only participates in SIP messages. Once the call is established, the phones send their voice traffic directly to each other without involving the proxy. SIP proxies offload tasks and simplify the implementation of end station telephones.
- The *B2BUA* is a SIP-based logical entity that receives and processes INVITE messages as a SIP User Agent Server (UAS). It also acts as a SIP User Agent Client (UAC) that determines how the request should be answered and how to initiate outbound calls. Unlike SIP proxy mode, the B2BUA maintains the call state and participates in all call requests.
- A *stateless* proxy forwards every request it receives and discards information about the request message once the message has been forwarded.

CLI session

The following CLI session sets the SIP operation mode to “proxy.”

```
NNOS-E> config vsp
config vsp> config default-session-config
config default-session-config> config sip-settings
config sip-settings> set mode proxy
```

CLI session

The following CLI session enables the SIP protocol on the specified IP interface, specifies the TCP, UDP and TLS ports to use when listening for SIP messages, and includes a certificate from an authorized certificate authority (CA).

```
NNOS-E> config cluster
config cluster> config box 1
config box 1> config interface eth0
config interface eth0> config ip boston1
Creating ‘ip boston1’
config ip boston1> config sip
config sip> set admin enabled
config sip> set nat-translation enabled
config sip> set nat-add-received-from enabled
config sip> set udp-port 5060
config sip> set tcp-port 5060
config sip> set tls-port 5061
config sip> set certificate vsp tls certificate cxc.net.com
```

For information on requesting and importing certificates from a CA, as well as configuring certificates on the AA-SBC device, refer to Chapter 6, “Managing certificates and keys with LCS and OCS.”

Load balancing across AA-SBC interfaces

Load balancing of SIP processing across interfaces requires both headed and backing interfaces.

The *headend* interface is the central distribution point. It does not perform SIP processing, it only forwards the calls to its configured backing interfaces. When you configure a SIP phone, configure the phone directly to the headend interface. To configure an IP interface as a headend interface, configure the **sip** object with backing interfaces. An interface is considered a headend interface if it has configured backing interfaces.

The *backing-interfaces* are identified within this **sip** object. In the **backing-interface** property, you reference previously configured IP interfaces. The backing interface is the location at which the AA-SBC terminates TCP and TLS connections (and where UDP transport messages arrive) and handles SIP processing. AA-SBC uses round-robin load-balancing to distribute message across the configured backing interfaces.

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, the interfaces 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.

To verify your configuration, first ensure that all SIP properties match. From the CLI at the AA-SBC device 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.

CLI session

```
NNOS-E> config cluster
config cluster> config box 1
config box 1> config interface eth0
config interface eth0> config ip boston1
config ip boston1> config sip
config sip> set admin enabled
config sip> set nat-translation enabled
config sip> set udp-port 5060
config sip> set tcp-port 5060
config sip> set tls-port 5061
config sip> set certificate “vsp tls certificate cxc.companyA.com”
config sip> set backing-interface “cluster box 1 interface eth0 ip
 backing1”
config sip> set backing-interface “cluster box 1 interface eth1 ip
 backing2”
config sip> set backing-interface “cluster box 2 interface eth0 ip
```

Configuring media port pools

The **media-ports** object defines the ports and port ranges to assign to media streams on an Ethernet interface, such as NAT, media anchoring, and media recording.

CLI session

The following CLI session enables the media-ports object, sets the starting port number, sets the total number of ports available for media streams, and enables the monitoring of idle ports (so that no traffic is sent to idle ports that are part of the media pool).

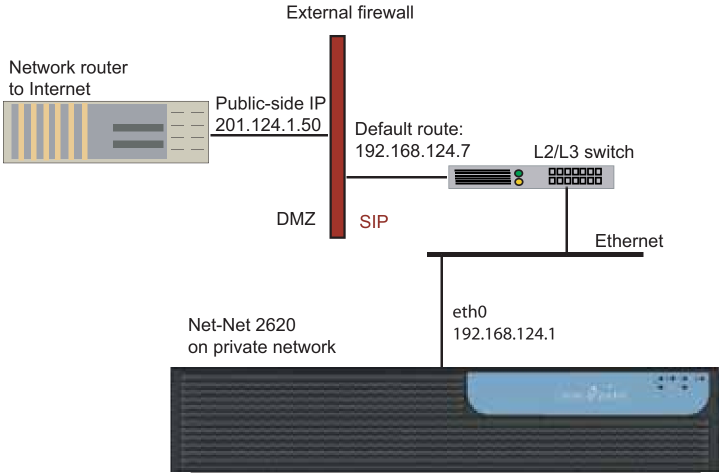
```
NNOS-E> config cluster
config cluster> config box 1
config box 1> config interface eth0
config interface eth0> config ip boston1
Creating ‘ip boston1’
config ip boston1> config media-ports
config media-ports> set admin enabled
config media-ports> set base-port 20000
config media-ports> set count 5000
config media-ports> set idle-monitor enabled
```

Configuring the external firewalls

The **near-side-nat** object allows you to configure the AA-SBC device to perform Network Address Translation (NAT) on SIP traffic that traverses the enterprise firewall between the AA-SBC device and the Internet. By configuring the IP address of the public-facing interface on the enterprise firewall, AA-SBC produces a contact header that replaces enterprise private IP addresses with the public-facing firewall address.

NAT, defined in RFC 1631, *The IP Network Address Translator*, ensures that internal private network addresses are rewritten so that they appear to come from the designated external network firewall address. AA-SBC modifies outgoing packets so that the return address is a valid Internet host (the external firewall). The firewall then changes the destination address on incoming packets to the AA-SBC device’s private address.

SIP traffic that matches the configured UPD and TCP port ranges will use NAT so that only the public-side IP address can be observed by remote SIP users across the Internet. The following image illustrates a sample network with showing the public-side IP address 204.124.1.50.



CLI session

The following CLI session configures the external firewall public IP address to 201.124.1.50 and sets the UDP and TCP port ranges over which to listen for SIP messages and IP address replacement.

```
NNOS-E> config cluster
config cluster> config box 1
config box 1> config interface eth0
config interface eth#> config ip boston1
config ip boston1> config near-side-nat 201.124.1.50
Creating ‘near-side-nat 201.124.1.50’
config near-side-nat 201.124.1.50> set admin enabled
config near-side-nat 201.124.1.50> set public-ip 201.124.1.50
config near-side-nat 201.124.1.50> set udp-range 5060 1
config near-side-nat 201.124.1.50> set tcp-range 5060 2
```

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, as described in this chapter.

In addition, you may configure the NAT pool addresses within this object, typically UDP ports 20000 through 30000.

Configuring the STUN, TURN, and ICE protocols

The AA-SBC, as a STUN server, uses three protocols that operate together to handle SIP signaling and media traversal across NAT routers and firewalls. These protocols are:

- STUN — Simple Traversal of User Datagram Protocol Through Network Address Translators
- TURN — Traversal Using Relay NAT
- ICE — Interactive Connectivity Establishment

The AA-SBC device implements draft-ietf-behave-rfc3489bis-04 (for STUN, in addition to RFC3489), and draft-ietf-behave-turn-01, both released in July 2006.

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-04—Simple Traversal Underneath Network Address Translators (NAT) (STUN)
- draft-ietf-behave-turn-01—Obtaining Relay Addresses from Simple Traversal of UDP Through NAT (STUN)

STUN

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. A STUN server and receives and transmits UDP messages over UDP port 3478 (default). The STUN protocol helps prevent NAT-associated network application failures by transmitting exploratory STUN messages over UDP between the server and clients.

STUN identifies the public side NAT details by inspecting exploratory STUN messages that arrive at the STUN server. The STUN-enabled client sends an exploratory message to the STUN server to determine the transmit and receive UDP port to use. The 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.

For complete information on STUN, refer to *RFC 3489 -STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs).*

Traversal Using Relay NAT (TURN)

Depending on the network topology and the NAT implementation, IP addresses obtained by STUN may not be usable by all peers. A client must be able to obtain a publicly visible transport address that can receive media from any peer that can send packets to the public Internet. This is done by relaying data though a server that resides on the public Internet.

The Traversal Using Relay NAT (TURN) protocol allows a client to obtain a transport address from a relay,

1. to send traffic to the peer through that address, and
2. to receive all traffic sent to that address by the peer.

The **relay-interface** property specifies the interface over which the SIP client receives public visibility, as well as the interface from which the AA-SBC device allocates TURN relay ports. This interface must have **media-ports** enabled and a port pool range defined.

Interactive Connectivity Establishment (ICE)

Both STUN and TURN work in conjunction with the Interactive Connectivity Establishment (ICE) protocol to determine what type of NAT firewalls exist between SIP clients and to determine a set of “candidate” transport addresses by which they are able to establish contact. STUN and TURN are sometimes used by ICE.

An ICE-enabled client (the initiator) uses STUN and TURN, and a locally configured policy, to determine a prioritized list of candidate addresses before sending this list to the responder client in the SDP portion of a SIP message. When the responder client receives this message, it performs a “connectivity check” on each candidate address by sending a STUN request to that address and waiting for a reply. The highest priority candidate to pass the connectivity check is then used for the actual connection.

The AA-SBC device does not require any ICE-specific configuration.

Sample configuration

For STUN to operate properly, follow these rules when configuring STUN servers:

- Create STUN server instances in pairs (for compliance with the RFC 3489).
- 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 of each STUN server instance must point at the IP address of the other STUN server 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.

The following CLI session configures a STUN server on the AA-SBC device.

CLI session

```
NNOS-E> config cluster
config cluster> config box 1
config box 1> config interface eth0
config interface eth0> config ip a
config ip a> config stun-server
config stun-server> set admin enabled
config stun-server> set stun-auth-level allow
config stun-server> set port 3478
config stun-server> set allow-turn enabled
config stun-server> set relay-interface “cluster box 1 interface
   eth1 ip abc”
config stun-server> set secondary-interface “cluster box 1 interface
 eth0 ip b”

NNOS-E> config cluster
config cluster> config box 1
config box 1> config interface eth0
config interface eth0> config ip b
config ip b> config stun-server
config stun-server> set admin enabled
config stun-server> set stun-auth-level allow
config stun-server> set port 3478
config stun-server> set allow-turn enabled
config stun-server> set relay-interface “cluster box 1 interface
   eth1 ip abc”
```

```
config stun-server> set secondary-interface “cluster box 1 interface
 eth0 ip a”
```

For more information, and for information on setting all stun-server properties, refer to the *Net-Net OS-E – Objects and Properties Reference*.

Configuring kernel filtering

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. In this way, 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.

CLI session

The following CLI session creates and enables a deny rule named *evil-badguy* from source IP address 215.200.40.8, source port 56, over UDP.

```
NNOS-E> config cluster
config cluster> config box 1
config box 1> config interface eth0
config interface eth0> config ip boston1
config ip boston1> config kernel-filter
config kernel-filter> config deny-rule evil-badguy
Creating ‘deny-rule evil-badguy’
config deny-rule evil-badguy> set admin enabled
config deny-rule evil-badguy> set source-address/mask 215.200.40.8/24
config deny-rule evil-badguy> set source-port 56
config deny-rule evil-badguy> set protocol udp
```

Configuring messaging

Messaging is the mechanism from which the AA-SBC device communicates with other systems in the cluster. Messaging sets up a listening socket on an interface, enabling the interface to receive messaging traffic and participate in clustering and media partnering.

In a cluster, the master looks through the configurations of all AA-SBC devices to find out which interface is used 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.

In media partnering, you configure a specific IP address (on a different box) as a partner. On the box that owns that IP address, you need to configure and enable messaging for media partnering to operate.

CLI session

The following CLI session configures messaging on box 1, interface eth0.

```
NNOS-E> config cluster
config cluster> config box 1
config box 1> config interface eth0
config interface eth0> config ip boston1
config ip boston1> config messaging
config messaging> set admin enabled
config messaging> set certificate vsp tls certificate name
config messaging> set port 13002
config messaging> set protocol tls
```

3. Enabling AA-SBC services

About this chapter

The chapter describes the services that you can enable on AA-SBC platforms.

Enabling services on the AA-SBC master

There are administrative services available on the AA-SBC master that are enabled by default. These master services are:

- Cluster-master services
- Directory services
- Accounting services
- Authentication services
- AA-SBC database
- Registration services
- Server load
- Call failover (signaling and media)
- Load-balancing
- File-mirror
- Route Server
- Sampling
- Third-party-call-control (3PCC)

If you are not using any of these services, you can globally disable them to conserve memory and system resources on the AA-SBC master.

Cluster-master services

The **cluster-master** services object configures the AA-SBC platform that maintains the master configuration for the cluster. The master is responsible for providing configuration changes and updates to other devices in the cluster. If a different device becomes the cluster-master during a failover, this device then sends out its configuration to the other devices in the cluster.

CLI session

```
NNOS-E> config
config> config master-services
config master-services> config cluster-master
config cluster-master> set admin enabled
config cluster-master> set host-box cluster box 2
config cluster-master> set host-box cluster box 1
```

Directory services

When enabled, directory services allows the AA-SBC master to use enterprise (or corporate) directories that contain the identities of SIP users who are authorized to access the SIP enterprise communications servers.

You can configure the SIP communications servers and associated user and directory attributes under the VSP (Virtual System Partition), enterprise configuration object.

CLI session

The following session enables directory services on the AA-SBC master.

```
NNOS-E> config master-services
config master-services> config directory
config directory> set admin enabled
config directory> set host-box cluster box 1
config directory> set host-box cluster box 2
config directory> set group 1
```

Accounting services

When enabled, accounting services supports RADIUS accounting, system logging (syslog), DIAMETER protocol services, the accounting database, archiving, and the accounting file-system.

You can configure one or more of these accounting mechanisms for capturing AA-SBC network accounting activity and SIP call detail records under the VSP (Virtual System Partition) configuration object.

CLI session

The following session enables the AA-SBC global accounting services on the master.

```
NNOS-E> config master-services
config master-services> config accounting
config accounting> set admin enabled
config accounting> set host-box cluster box 3
config accounting> set host-box cluster box 1
config accounting> set group 1
```

Authentication services

Authentication services enables or disables all authentication functions on the AA-SBC device, such as RADIUS and local user profiles. If authentication is disabled, you can still configure the authentication services, but the services do not become active until you enable this master service.

CLI session

The following session enables AA-SBC authentication services.

```
NNOS-E> config master-services
config master-services> config authentication
config authentication> set admin enabled
config authentication> set host-box cluster box 3
config authentication> set host-box cluster box 1
config authentication>
```

AA-SBC database

The master-services **database** object allows you to configure maintenance and other settings for the AA-SBC system database. The database is the local repository for call accounting records and media files.

CLI session

The following session enables AA-SBC database maintenance and sets the local maintenance time at 6 a.m. daily.

```
NNOS-E> config master-services
config master-services> config database
config database> set admin enabled
config database> set maintenance time-of-day 06:00
```

Registration services

Enabling the registration service allows AA-SBC to accept SIP REGISTER requests in behalf of other SIP servers (called *registrar peers*) that reside in other domains.

SIP registrars each maintain addresses of record bindings and contact addresses for their own domains in a location service database. Registrar peers exchange these location records with each other during REGISTER sessions. When AA-SBC accepts a REGISTER request from a foreign domain, for example, AA-SBC installs a binding in its location database so that subsequent SIP messages are forwarded to the SIP client.

For detailed information on registration and location services, refer to the *Net-Net OS-E – Session Services Configuration Guide*.

CLI session

The following session enables the registration services on the AA-SBC master.

```
NNOS-E> config master-services
config master-services> config registration
config registration> set admin enabled
config registration> set host-box cluster box 1
config registration> set host-box cluster box 2
```

Server load

The master-services **server-load** object 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.)

CLI session

The following session enables the server load functionality on the AA-SBC master.

```
NNOS-E> config master-services
config master-services> config server-load
config server-load> set admin enabled
config server-load> set host-box “cluster box 2”
config server-load> set host-box “cluster box 3”
```

Call failover (signaling and media)

The master-services **call-failover** object configures failover for both the media and signaling streams across a AA-SBC cluster. Enabling **call-failover** ensures that there is an active copy of the database on another box in the cluster in the event of a failure. The first **host-box** property defines the primary AA-SBC device. Configure backup boxes in the event of primary failure by re-executing the **host-box** property.

CLI session

The following session enables call-failover of the media and signaling streams.

```
NNOS-E> config master-services
config master-services> config call-failover
config call-failover> set admin enabled
config call-failover> set host-box cluster box 1
config call-failover> set host-box cluster box 2
```

The call must be connected at the SIP level for signaling failover to succeed. States prior to the “connected” state are not maintained in the cluster-wide state table. For TCP and TLS connections, the user agent (UA) must reestablish the connection after the failover, since TCP and TLS are connection-oriented protocols that do not maintain state information. If TLS is used, the appropriate certificate must be loaded on both devices in the cluster.

Accurate call logs are recorded at the end of the call. However, if the AA-SBC device maintaining the call log database fails over to the other AA-SBC device in the cluster, call information will not be recorded.

Use the AA-SBC status **show signaling-sessions** command to view cluster-wide signaling state information.

```
NNOS-E> show signaling-sessions

  session-id: 342946641025485482
  fromURI: <sip:1234@dial-plan.com>
  toURI: <sip:5678@dial-plan.com>
  inLegCallID: 3c2a54ca1fbd-7intxouoq8zo@172-30-0-176
  inLegFromTag: xqkhmbwmiv
  inLegToTag: b432a8c0-13c4-454a1124-102dd42a-164adf67
```

```
outLegCallID:
  CXC-279-61b29378-b432a8c0-13c4-454a1124-102dd42b-7023adbd@dial-pla
  n.com
outLegFromTag: b432a8c0-13c4-454a1124-102dd42b-749c0b03
outLegToTag: 152jkzyt73
origInFromURI:
origInToURI:
origOutFromURI:
origOutToURI:
vthreadID: 278
initialMethod: 0
Box: 0.0.0.0
```

Load-balancing

The master-services **load-balancing** object configures AA-SBC devices to host the load-balancing master service. These devices (boxes) are responsible for keeping the rule database up to date. They do not need to be the same devices that host the head-end interfaces, although it is common to do so. (You can, for example, configure devices in the cluster that only serve as host devices without any head-end interfaces or backing interfaces.)

For more information on the load-balancing object, refer to the *Net-Net OS-E – Objects and Properties Reference*. For more information on configuring load-balancing across AA-SBC Chapter 2: Enabling Networks and Protocols.

CLI session

The following CLI session enables load balancing on the master, specifies box 1 as the master box on which the rule database runs (subsequent host boxes 2 and 3 serve as backup) and associates the load balancing service with preconfigured VRRP group 1.

```
NNOS-E> config master-services
config master-services> config load balancing
config load-balancing> set host-box cluster box 1
config load-balancing> set host-box cluster box 2
config load-balancing> set host-box cluster box 3
config load-balancing> set group 1
```

File-mirror

The master-services **file-mirror** object sets all participating AA-SBC devices to share particular files (the types of files shared are preset in the operating system), such as media recordings, log files, etc. The file-mirror master service distributes files to all AA-SBC 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's database. The master database then sends a message to all backup databases indicating a change and updating the backup.
2. 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).
3. The backup then pulls the missing file(s) from the master's 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 box 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. In this way, 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.

CLI session

The following CLI session enables file-mirroring on the AA-SBC device. This CLI session also specifies box 1 as the master box on which file mirroring is run, associates the file mirroring process with VRRP group 1, and identifies *cxc_common/mirror1* as the location to which AA-SBC writes the files.

```
NNOS-E> config master-services
config master-services> config file-mirror
config file-mirror> set admin enabled
config file-mirror> set group 1
config file-mirror> set host-box cluster box 1
config file-mirror> set file-mirror-directory /cxc_common/mirror1
```

Route Server

The master-services **route-server object** sets the route-server (RS) master service, which manages the server process. The master service handles requests from local or remote AA-SBC devices for 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) RS 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 RS 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.

Note: This section describes how to configure intracluster routing. For more information about configuring intercluster routing, see *Net-Net OS-E – Objects and Properties Reference*

Note that because AA-SBC propagates RS 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 LCR import client, a web application that imports routes into the database.

CLI Session

When two or more AA-SBC devices are within a cluster and the RS server resides in the cluster, you can use intracluster RS. In that case, the RS lookup process is handled by the AA-SBC device within the cluster running this master service.

To use intracluster routing, you must configure the following:

1. Set the primary and backup AA-SBC devices that will host the least-cost-routing master.
2. Enable the lookup destination by setting the **mode** property of the session configuration **authorization** object to **local**.

The following CLI session configures intracluster routing on the AA-SBC master, specifying box 1 as the primary box, and box 2 and box 3 as backup boxes. It also sets the authorization mode to local.

```
NNOS-E> config master-services
config master-services> config least-cost-routing
config least-cost-routing> set admin enabled
config least-cost-routing> set host-box cluster box 1
config least-cost-routing> set host-box cluster box 2
config least-cost-routing> set host-box cluster box 3
config least-cost-routing> return
config master-services> return
config>

config> config vsp
config vsp> config default-session-config
config default-session-config> config authorization
config authorization> set mode local
config authorization> return
config default-session-config> return
config vsp> return
config>
```

Sampling

The master-services **sampling** object 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. See *Net-Net OS-E – Objects and Properties Reference* for more information on configuring these 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 select a status provider underneath Trends in the Status tab of 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, and are not using this feature, disable it. Otherwise, polling data is continuously written to the status database.

CLI Session

The following CLI session enables sampling services on the AA-SBC master:

```
NNOS-E> config master-services
config master-services> config sampling
config sampling> set admin enabled
config sampling> set host-box cluster box 1
config sampling> set host-box cluster box 2
config sampling> set host-box cluster box 3
config sampling> set group 1
config sampling> return
config master-services> return
config>
```

Third-party-call-control (3PCC)

The master-services **3pcc** (third-party-call-control) object configures call control, allowing the 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 AA-SBC. 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. In this way, phone control can be integrated, for example, into Microsoft Office applications via 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, 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.

CLI Session

The following CLI session enables third-party-call-control services on the AA-SBC master:

```
NNOS-E> config master-services
config master-services> config 3pcc
config 3pcc> set admin enabled
config 3pcc> set host-box cluster box 1
config 3pcc> set host-box cluster box 2
config 3pcc> set host-box cluster box 3
config 3pcc> set group 1
config 3pcc> return
config master-services> return
config>
```

Enabling event logging services

The AA-SBC event logger allows you to configure how event messages are filtered and captured. You can direct event messages to a remote syslog server (by IP address), to a named event log file stored on the AA-SBC platform, or to the local AA-SBC database.

CLI session

The following session configures the event logger to direct event messages to a remote syslog server.

```
NNOS-E> config services
config services> config event-log
config event-log> config syslog 192.168.124.89
```

CLI session

The following session configures the event logger to direct event messages to a named file and sets the event log operational parameters: direct all messages to the file, limit the event log file size to 20 Mbytes, and set the maximum number of event log files to create when log files reaches the maximum size in megabytes.

```
NNOS-E> config services
config services> config event-log
config event-log> config file eventfile1
config file eventfile1> set admin enabled
config file eventfile1> set filter all error
config file eventfile1> set size 20
config file eventfile1> set count 5
```

CLI session

The following session configures the event logger to direct event messages to the local AA-SBC database and sets the event log operational parameters: direct only SIP messages to the local database, and set the maximum number of days over which event messages are logged to the local database before the database is cleared and restarted.

```
NNOS-E> config services
config services> config event-log
config event-log> config local-database
config local-database> set admin enabled
config local-database> set filter sip error
config local-database> set history 50
```

Configuring threshold monitors

The **services/monitors** configuration object allows you to monitor the following statistics and thresholds for logging and SNMP trap generation:

- CPU usage
- Memory usage
- TLS connections statistics

Polling intervals are in minutes, memory and CPU usage in percent, and TLS connections and failures in actual numbers. At the specified polling interval(s), AA-SBC checks memory and CPU usage, and TLS statistics. If a parameter setting is exceeded, AA-SBC logs an event and an SNMP trap.

```
NNOS-E> config services
config services> config monitors
config monitors> config monitor usage
Creating ‘monitor usage’
config monitor usage> set interval 60
config monitor usage> set parameter cpu-usage 90
config monitor usage> set parameter memory-usage 95
config monitor usage> return
config monitors> config monitor tls
Creating 'monitor tls'
config monitor tls> set interval 30
config monitor tls> set parameter tls-connections 1000
config monitor tls> set parameter tls-failures 10
```

Configuring data and archiving locations

The **services/data-locations** configuration object allows you to specify the directory and path locations on the AA-SBC device where you are to save certain types of information. This information includes:

- RTP media (for call recording). Use the **rtp-recorded** property to select a location on the system disk for local archiving of call detail records and call recordings.
- RTP mixed (for playback of recorded calls). Use the **rtp-mixed** property to set the location for playback of recorded calls.
- File transfer. Use the **file-transfer-recorded** property to set the location for file transfer records.
- Log files. Use the **log** property to set the location for log files.

If you choose not to create specific locations for saved files, AA-SBC provides default directory path locations. For example, the directory path */cxc_common* on hard-drive-1 is the default location for recorded RTP files and file transfers. You can display the default directory file paths using the **show** command.

CLI session

```
config> config services data-locations
config data-locations> show

services
 data-locations
  rtp-recorded[1] /cxc_common/rtp_recorded
  rtp-recorded[2] /cxc/recorded
  rtp-mixed[1] /cxc_common/rtp_mixed
  rtp-mixed[2] /cxc/mixed
  rtp-mixed[3] /cxc/admin/archives
  file-transfer-recorded[1] /cxc_common/ft_recorded
  file-transfer-recorded[2] /cxc/recorded
  log /cxc_common/log
```

The following CLI session changes the default logging path from */cxc_common/log* to */cxc/admin/logfiles*.

CLI session

```
config> config services data-locations
config data-locations> set log /cxc/admin/logfiles
```

The following CLI session sets the location for “mixed” RTP files to the directory */cxc/admin/RTPmixed;* the location for storing file transfer records is set to */cxc/admin/FTrecords.*

CLI session

```
config> config services data-locations
config data-locations> set rtp-mixed /cxc/admin/RTPmixed
config data-locations> set file-transfer-recorded /cxc/admin/FTrecords
```

Configuring an external database

If you want to use a database other than the one that is provided with the AA-SBC platform, you can configure AA-SBC to use an external database to store event logs, call detail records, and other accounting data. Depending on your network remote SQL server databases, for example, can provide large storage and resource capabilities.

To configure an external database, you will need the Open Database Connectivity (ODBC) driver name associated with the database, as well the user name and secret tags (and password) needed for AA-SBC to access the database. Consult your database administrator for this information before configuring the remote database on the AA-SBC platform.

The following CLI session configures the database driver named “My SQL Server” and sets the username, secret-tag, and password/password confirmation for this database.

CLI session

```
config> config services database external
config database external> set driver “My SQL Server”
config database external> set username cxc
config database external> set secret-tag 123
password: *********
 confirm: *********
config database external>
```

The following CLI session configures the event log to direct *snmp* events with the severity *warning* to the SQL database named *corpDatabase* for a period of 150 days. AA-SBC automatically associates the external database name to the services/database configuration.

CLI session

```
config services> config event-log
config event-log> config external-database corpDatabase
config corpDatabase> set admin enabled
config corpDatabase> set filter snmp warning
config corpDatabase> set history 150
```

For more information on the services/database object, refer to the *Net-Net OS-E – Objects and Properties Reference*.

Setting AA-SBC disk thresholds

The **storage-device** object allows you to set warning and failure thresholds for remaining disk space on AA-SBC hard drives. When a disk drive reaches the configured **fail-threshold** property setting, AA-SBC begins WRITE operations to the next available disk drive. Warning messages are logged (in minutes) whenever disk threshold settings are matched.

The **storage-device** object operates on all installed disk drives. If all disk drives match the configured thresholds, media call recording, file transfers, and log files will no longer be written to the AA-SBC disks.

➡ **Note:** Currently, AA-SBC platforms support two 250GB internal disk drives.

The following CLI session sets the fail thresholds for all installed disk drives. Writing to that disk fails when the remaining disk space drops to 20 GB.

CLI session

```
config> config services
config services> config storage-device
config storage-device> set fail-threshold 20000
```

Scheduling regularly performed tasks

AA-SBC can automatically perform tasks on a configured schedule. This means that you do not have to physically execute an action at a specific time; AA-SBC does it for you. Use the **set action ?** command to display the current list of tasks from which you can choose.

The following CLI session configures a directory reset for the *Boston1* enterprise directory at 3 pm.

CLI session

```
config> config services
config services> config tasks
config tasks> config task directoryReset
config task 1> set action ?
  archive         Run the archiving task for a given vsp
  directory-reset Reset an enterprise directory

config task 1> set action directory-reset
config task 1> set schedule time-of-day 15:00:00
```

Performing database maintenance

AA-SBC automatically runs a database maintenance script daily, at 3:00 A.M. This “normal” database maintenance purges (removes old files preventing AA-SBC disks from becoming too full), “vacuums” (reclaims unused disk space), reindexes, and analyzes the database. You can also selectively schedule periodic database maintenance or force database maintenance at any time.

Along with normal daily database maintenance, it is recommended that you perform a “vacuum-full” on the database monthly to reclaim unused disk space.

This section describes how to do the following database maintenance tasks:

- Set normal maintenance time-of-day.
- Schedule periodic database maintenance.
- Force manual database maintenance.
- Perform the database “vacuum-full” process (recommended monthly, in addition to normal maintenance).

Note: As a guideline, it is recommended that you perform database archiving more frequently than database maintenance. For example, archiving on a daily basis and performing maintenance every 7-days allows records in the database to age without the risk of removing records before those records are archived.

See the section in the chapter, “Enabling and configuring local archiving” for information.

Setting normal database maintenance time-of-day

AA-SBC automatically runs database maintenance daily, at 3:00 A.M.

If you want to change the actual time-of-day when AA-SBC runs normal database maintenance, use the **master-services** database object. If old records are found, AA-SBC purges those records from the database. Optionally, you can configure a time period in hours, such as every 3 hours, if you want run maintenance at multiple time periods during a 24-hour day.

CLI session

```
config> config master-services
config> config database
config database> set maintenance time-of-day 02:00:00
```

Verifying normal database maintenance

To verify that normal maintenance has successfully completed, and that a table has been vacuumed automatically, view the system log file. The log file should display a message similar to the one shown below:

```
2008-04-16T05:39:45+12:00[notice] 1:SIP[system] An automatic VACUUM
FULL was performed on database table SipMessage to reclaim 895300
unused pointers
```

Scheduling periodic database maintenance

The VSP **database** object allows you to configure the number of days to elapse before AA-SBC purges old records from the database. You can selectively configure the number of days for each of the following database records:

- accounting
- call details
- media
- file transfer
- instant messages

Whenever records in the database become older than the configured number of days, the next maintenance natively purges the old files. The following CLI session configures the number of days to elapse for each database record type before AA-SBC deletes the old records from the system disk.

CLI session

```
config> config vsp
config> config database
config database> set accounting-history 7 days
config database> set call-details 10 days
config database> set media-history 10 days
config database> set file-transfer-history 3 days
config database> set im-history 2 days
```

Forcing database maintenance

Use the **database-maintenance normal** command to run a specific database maintenance operation at any time. This forces a database cleanup of any old database entries if you did not previously configure the VSP database settings. Use the show **database-tables** command to display the database contents after the cleanup.

CLI session

```
NNOS-E> database-maintenance normal
```

```
Starting database maintenance as a background operation.

 -- this may take a very long time --

Please check database-maintenance-status for notification when this
operation is complete.
```

Performing database vacuum-full

The normal (daily) vacuum process attempts to reclaim any unused space in the database (this is analogous to a hard drive defragmentation process, but on the database files) without locking any of the tables.

The **database vacuum-full** process locks each table, one at a time, and reclaims all possible disk space. Note that a table lock prevents the table in question from being written to by an application; i.e. AA-SBC.

It is recommended to perform a **database vacuum-full** on a monthly basis by scheduling a “maintenance/outage window.” You should only run the process during a maintenance window because the process will lock tables, preventing them from being written to by AA-SBC. This can affect the ability of a DOS rule from being triggered, and at the same time, affecting call-logs, recording of accounting data, and any other data that is written to the database. (However, running **database vacuum-full** will not affect the ability of AA-SBC to pass sip/media traffic, accept/ delegate registrations, route calls, and perform other directly service-related tasks).

If your site is logging a large volume of data, you may wish to perform a **vacuum-full** on a more frequent (e.g. weekly) basis.

Note the following **database vacuum-full** implementation tips:

- You perform a vacuum-full on the entire database (global) using the **database vacuum-full system** command.
- You vacuum a specific table using the **database vacuum-full system <table-name>** command. For example, you may wish to use this process if AA-SBC logs a message stating that a specific table needs to be vacuumed.

Performing other database maintenance tasks

You use the VSP **database** object to perform other database maintenance tasks, as described below:

- **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 SQL's 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 (versus more extensive **vacuum-full process** during which the table is not available for read/write operations during the process).
- **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 AA-SBC Release 3.2 and later.
- **initialize**—Deletes all data and reinitializes the database.
- **snapshot**—Captures an archive of the database at a certain point(s) in time.

Refer to the *Net-Net OS-E – Objects and Properties Reference*for full description of how to use each of these database objects.

Managing Net-Net-OS-E database size

This section describes ways to manage the size of the AA-SBC database. That is, it describes ways to reduce the amount of data that is being written to the database. You may wish to try one of these procedures if your database is growing too large, or if it is responding too slowly:

- Disable the logging of REGISTER messages.
- Configure a policy to prevent logging of NOTIFY messages.

Note: Backup the current AA-SBC configuration before attempting any of the procedures described in this section.

Disabling REGISTER message logging

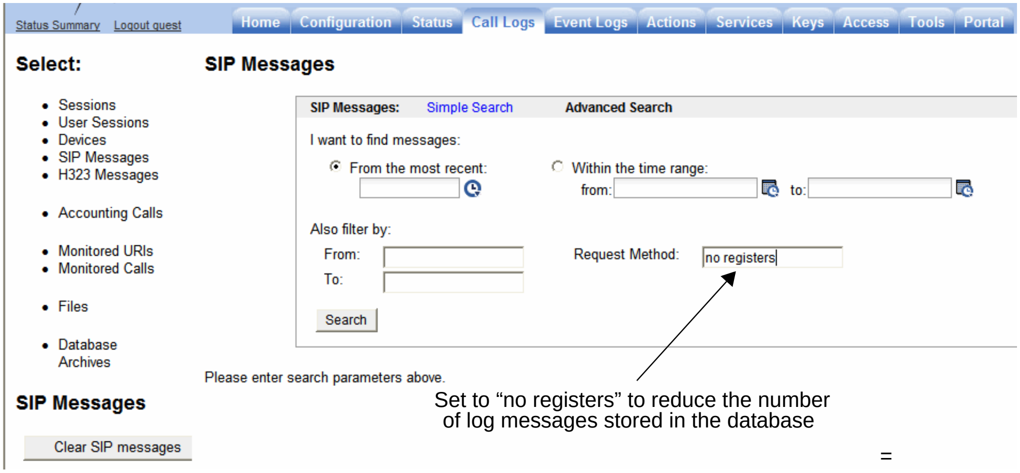
To disable the logging of REGISTER messages in order to reduce the amount of data store in the database, do the following steps:

1. From the AA-SBC Management System, select **vsp->default-session-config->log-alert**.
2. Set **message-logging** to **no-registers**, then click **Set** to save your changes. This change will take effect immediately.
3. Repeat Step 1 for any **session-configs** that have a **log-alert** configured (e.g. in session-config-pool entries, policies, dial/registration-plans, etc.).

To verify that REGISTER messages are no longer being logged, do the following steps:

1. From theAA-SBC Management System, click on the **Call Logs** tab.
2. From the left side of the window, click **SIP Messages**.
3. Click on **Advanced Search**.
4. Enter in **REGISTER** in the **Request Message** field.

   AA-SBC searches for messages of type REGISTER. None should be found.



I want to find messages:

● From the most recent:

○ Within the time range: from: to:

Also filter by:

From:

To:

Request Method: no registers

Please enter search parameters above.

Set to “no registers” to reduce the number of log messages stored in the database

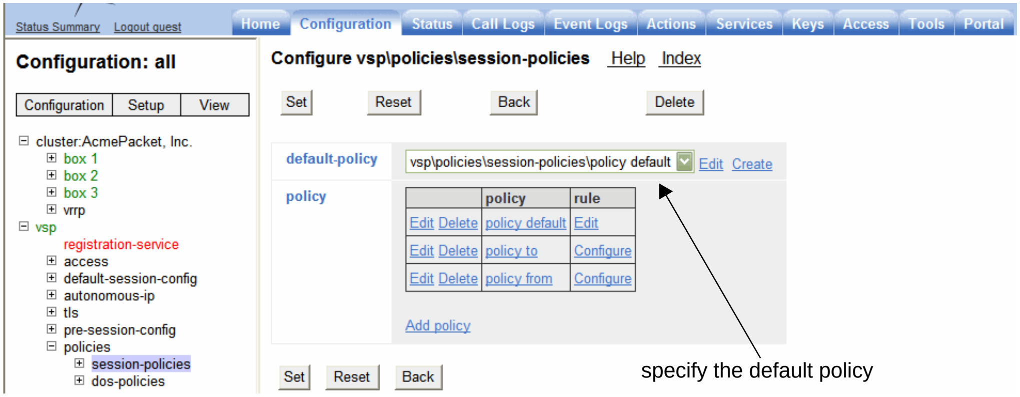
Preventing NOTIFY message logging

If you want to further reduce the amount of data that is being logged to the database, you can configure a policy to prevent logging of specific message types. For example, you may want to prevent the logging of NOTIFY messages that are received from phones (i.e. being received on the public IP interface). These messages are often used as “keep-alive” messages from the end device.

To configure a policy to prevent logging of NOTIFY messages from phones, do the following steps:

1. From the AA-SBC Management System, select **vsp->policies**.
2. Scroll to **session-policies**, then click **Add policy**.
3. Name the new policy “default” and then click **Create**.

   If there is already a default-policy configured under **vsp->policies** skip to step 4, keeping in mind that in this example the default-policy is named “default.”



default-policy: vsp\policies\session-policies\policy default

policy

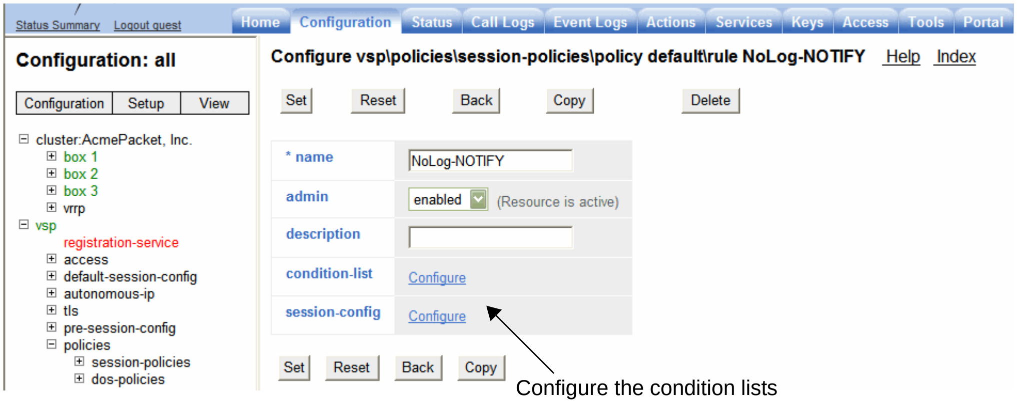
	policy	rule
Edit Delete	policy default	Edit
Edit Delete	policy to	Configure
Edit Delete	policy from	Configure

specify the default policy

4. Specify the “default” policy as your default-policy under **vsp->policies-> session-policies.**
5. Under **vsp->policies-> session-policies->policy default**, add a new rule. Name the new rule something obvious, for example, “NoLog-NOTIFY.”
6. Under vsp->policies-> session-policies->policy default->rule-> NoLog-NOTIFY, configure the condition-list as follows:

   — Set the **default operation** to **AND**

   — Set the **mode** to **evaluate**



⊟ cluster:AcmePacket, Inc.
- ⊞ box 1
- ⊞ box 2
- ⊞ box 3
- ⊞ vrrp

⊟ vsp
- registration-service
- ⊞ access
- ⊞ default-session-config
- ⊞ autonomous-ip
- ⊞ tls
- ⊞ pre-session-config
- ⊟ policies
  - ⊞ session-policies
  - ⊞ dos-policies

Configure vsp\policies\session-policies\policy default\rule NoLog-NOTIFY

* name	NoLog-NOTIFY
admin	enabled (Resource is active)
description	
condition-list	Configure
session-config	Configure

Configure the condition lists

7. Under **vsp->policies-> session-policies->policy default->rule-> NoLog-NOTIFY->condition list**, add a **sip-message-condition**, as follows:

   — Set an **attribute** of **request-method**

   — Set the **match** to **match**

   — For the **request-method**, select **NOTIFY** (You could also select other SIP message types.)

8. Under **vsp->policies-> session-policies->policy default->rule-> NoLog-NOTIFY->condition list**, add a **sip-message-condition,** as follows:

   — Set an **attribute** of **local-ip**

   — Set the **match** to **match**

   — For the **local-ip**, enter the IP Address of the AA-SBC public interface, including the “slash” subnet mask notation. For example: 1.1.1.1/32.

9. Under **vsp->policies->session-policies->policy default->rule-> NoLog-NOTIFY->condition list**, add a **sip-message-condition**, as follows:

   — Set an **attribute** of **direction**

   — Set the **match** to **match**

   — For the **value**, select **RX**

10. Under **vsp->policies-> session-policies->policy default->rule-> NoLog-NOTIFY**, create a **session-config** container.

11. **Under vsp->policies-> session-policies->policy default->rule-> NoLog-NOTIFY->session-config**, configure a **log-alert** container, then set **message-logging** to **disabled**.

To check to see if the rule is being enforced, perform a “show rules” from the CLI. For example:

```
NNOS-E> show rules
name: policy default/rule NOTIFY
admin: enabled
evaluations: 10008082
successes: 8336316

NNOS-E> show rules
name: policy default/rule NOTIFY
admin: enabled
evaluations: 10008127
successes: 8336356
```

If the number of “successes” is increasing, then the condition-list “entry criteria” are causing SIP messages to be affected by the rule’s session-config.

Backing up the database

The **database-backup backup** command allows you to create a backup file of the database, and save it to AA-SBC.

Note: Performing the database backup procedure increases the load on AA-SBC, slowing down the device. Therefore, it is recommended to perform this task for debugging purposes only.

The database backup file is saved in /cxc/pg_dump/*name*, where *name* is the file name that you specify. When you enter the name for the backup database file, make certain to specify a path that begins with /cxc/pg_dump/.

For example, /cxc/pg_dump/database1 is correct. However, if you specify /cxc/database1, 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 filename at the command line	Get an archive of this type
DBbackup	DBbackup.bz2
DBbackup.gz	DBbackup.gz

To create a database backup file and store it on AA-SBC, perform the following steps:

1. Use the **show mounts** command and **shell** command to verify that you have enough storage space on the disk (preferably /mnt/hd2), as shown in the following sample CLI session:

```
NNOS-E> show mounts
device       device-name   mount-point   filesystem    disk-size  percent-free
------       -----------   -----------   ----------    ---------  ------------
cdrom        /dev/cdrom                                0          0
usb          /dev/usb1                                 0          0
hard-drive-1 /dev/root     /             reiserfs      234448     96
hard-drive-2 /dev/sdb                                  0          0
hard-drive-3

NNOS-E> shell
bash-3.00# du -sh /var/lib/pgsql
296M    /var/lib/pgsql
bash-3.00# exit
exit
```

2. Because this procedure is used most often as a debugging tool, the /cxc/pg_dump directory is not present during the initial AA-SBC installation. Therefore, you must create it using the **mkdir** command as shown in the following CLI session:

CLI session

```
NNOS-E> shell
bash-3.00# mkdir pg_dump
bash-3.00# exit
exit
NNOS-E>
```

3. Execute the **database-backup backup** command, specifying a filename for the database backup file.

For example, the following CLI session creates a database backup file named **DBbackup.bz2** where **system** is the system database where call logs and accounting records are stored:

CLI session

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

Restoring a database

Use the **database-backup restore** command to restore a saved database backup file from the /cxc/pg_dump directory to the AA-SBC device.

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.

The following CLI session restores the backup file *backup.bz2*.

CLI session

```
NNOS-E> database-backup restore system /cxc/pg_dump/backup
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.
```

Enabling and configuring local archiving

Local archiving allows you to store call accounting records and media files at regular intervals on the AA-SBC platform before the records are removed by the database maintenance interval, as described in the previous section. Other archiving options “push” the data to alternate locations.

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 enable local archiving using the **vsp\accounting\archiving object**. In addition, you must configure a server in one of the archiving sub-objects for the archiving mechanism to work.

- **windows-share**
- **ftp-server**
- **smtp-server**
- **db-server**
- **local**

The following CLI session enables archiving on the ftp server named *ftp1*.

CLI session

```
NNOS-E> config vsp
config vsp> config accounting
config accounting> config archiving
config archiving> config ftp-server ftp1
config ftp-server ftp1> set admin enabled
config ftp-server ftp1> set username admin
config ftp-server ftp1> set password-tag xyz123abc
password: ************
confirm: ************
config ftp-server ftp1> set directory /archives
config ftp-server ftp1> set server 192.168.10.10
config ftp-server ftp1> set port 1998
config ftp-server ftp1> set timeout 100000
```

To locally archive on a scheduled basis, you need to schedule the archiving task.

CLI session

```
config> config services
config services> config tasks
config tasks> config task archive
config task archive> set action archive
config task archive> set schedule time-of-day 15:00:00
```

For more information on archiving and archiving to multiple server locations away from the AA-SBC platform, refer to Chapter 4, “Configuring AA-SBC accounting and archiving” and the *Net-Net OS-E – Objects and Properties Reference*.

4. Configuring AA-SBC accounting and archiving

About this chapter

The chapter describes the supported AA-SBC methods for capturing SIP call detail records (CDRs) and other accounting records associated with SIP sessions

Accounting system overview

AA-SBC uses industry-standard accounting targets where SIP call detail records are forwarded. The supported accounting targets are:

- RADIUS
- Database
- Syslog
- File system
- DIAMETER
- Archiving

Accounting records are written to directories on the file system, providing a large storage queue for call records as they are written. The accounting software then reads and distributes the call records to the configured accounting target destination(s).

In the event that an accounting target is unable, call records are automatically resent when the accounting target destination(s) become available and when all targets have been updated successfully. Use the **accounting reapply** action to resend call records in the file-system that met the date range to the target regardless if they previously were sent to the target successfully (or not).

The following directory structure store accounting records prior to their distribution to the various accounting targets.

```
/cxc/accounting/
    Subdirectories:   #
        Files: #-sessionid
```

Base directory — The root location on the AA-SBC device for storing CDRs, such as */cxc/accounting*.

Subdirectories — A series of numbered subdirectories each containing the number of files specified by accounting **subdirectory-size** property. The naming convention is *# - sequential value*.

Files — Each entry is a discrete CDR record. The naming convention is *# - sequential value* followed by the session identifier.

As the accounting software reads and processes files in the subdirectories, it creates, updates and deletes the following status markers:

- **complete** — Indicates that the directory has been fully populated and that all of the files in the directory have been successfully processed.
- **lastprocessed** — Indicates that the directory is currently being populated and that all of the files have been processed successfully.
- **pending** — Indicates that the accounting software has selected the directory for processing and that processing has not yet begun.
- **inprogress** — Indicates the files in the directory are currently being processed.
- **reapply** — Indicates that the directory is currently being evaluated by the **accounting reapply** action.

The **services\data-locations** object contains the **accounting-root-directory** property to specify the directory where accounting records will be placed prior to being sent to the various accounting targets. The default location is the */cxc_common/accounting* directory.

Configuring the accounting settings

General accounting settings are available under the **vsp\accounting** configuration object.

- **admin —** Enables or disables all configured accounting targets.

- **retention-period** — Specifies how many days the accounting records should be retained before being purged from the file system. The default setting is 7. The range is 0 to 21 days.
- **subdirectory-size** — Specifies the number of records to be recorded in each of the sub-directories. The default is 1000. The range is 100 to 2000.
- **purge-criteria** — Specifies he criteria to be used when deleting records from the file system. The **purge-always** setting indicated that records should be deleted even if they have not been saved to all of the defined enabled targets. The **purge-only-when-complete** setting indicates that even expired CDRs should be retained if they have not been sent to all of the defined targets.
- **report** — Creates a named CDR summary report containing the specified field, match, and category criteria.

The **accounting purge** action forces an immediate purge and clears all CDRs on the file system that are eligible for deletion.

The **accounting reapply** action accepts a date range and selected groups and marks qualifying records on the file system back to an unprocessed state. The records are picked up and reapplied (resubmitted) to the configured accounting targets. Use this action if CDR data is lost for a selected target and the data needs to be recovered. This action is limited to data within the current retention period.

The **show accounting-status** command provides a summary of current accounting and processing information for existing targets, including any target exceptions.

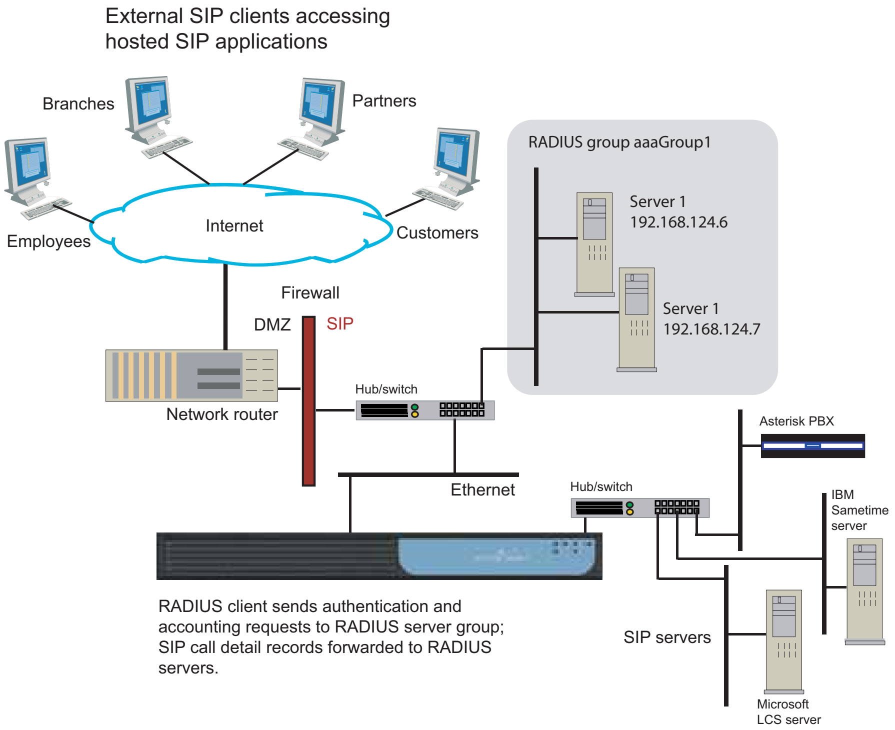
Configuring RADIUS groups

The Remote Authentication Dial In User Service (RADIUS) implementation allows the AA-SBC device to operate as a RADIUS client that directs SIP call detail records to a RADIUS accounting server. The RADIUS accounting server receives the accounting request and returns a response to the client indicating that it has successfully received the request.

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 one or more VSP configurations allows the AA-SBC device (the RADIUS client) to perform user authentication and forward accounting and SIP call detail records to 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.

Within a RADIUS group, you set the RADIUS authentication and accounting modes that you are using, the type of RADIUS accounting format, and whether the RADIUS 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.

The following image illustrates a sample network using a RADIUS accounting group.



CLI session

The following CLI session creates the RADIUS accounting group named **`aaaGroup1`** and sets the group operational properties.

```
NNOS-E> config vsp
config vsp> config radius-group aaaGroup1
Creating ‘radius-group aaaGroup1’
config radius-group aaaGroup1> set admin enabled
config radius-group aaaGroup1> set accounting-mode duplicate
config radius-group aaaGroup1> set authentication-mode failover 3
config radius-group aaaGroup1> set type Cisco
```

In this session, the authentication and accounting modes are RADIUS operational algorithms. 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. If you configure multiple authentication servers in the RADIUS group, the failover algorithm forwards authentication requests to secondary servers should the current authentication session fail. You can specify up to 256 failover attempts to other servers.

The default accounting method is cisco accounting, and the aaaGroup1 RADIUS group is a default group for all non-policy governed RADIUS requests between the AA-SBC device and the RADIUS servers.

Configuring the RADIUS servers

You can configure multiple RADIUS servers in the RADIUS group, and you identify each server using a unique number and IP address, authentication port, accounting port, and other operational settings.

CLI session

The following CLI session creates two numbered RADIUS servers and sets the operational properties for RADIUS requests and responses between the AA-SBC device and the RADIUS servers.

```
NNOS-E> config vsp
config vsp> config radius-group aaaGroup1
config radius-group aaaGroup1> config server 192.168.147.6
config server 192.168.147.6> set admin enabled
config server 192.168.147.6> set authentication-port 1800
config server 192.168.147.6> set accounting-port 1801
config server 192.168.147.6> set secret-tag abc123xyz
config server 192.168.147.6> set timeout 1500
config server 192.168.147.6> set retries 3
config server 192.168.147.6> set window 255
config server 192.168.147.6> set priority 2
config server 192.168.147.6> return

config vsp> config radius-group aaaGroup1
config radius-group aaaGroup1> config server 192.168.147.7
config server 192.168.147.7> set admin enabled
config server 192.168.147.7> set authentication-port 1800
config server 192.168.147.7> set accounting-port 1801
config server 192.168.147.7> set secret-tag abcXYZ123
config server 192.168.147.7> set timeout 1500
config server 192.168.147.7> set retries 3
config server 192.168.147.7> set window 255
config server 192.168.147.7> set priority 2
config server 192.168.147.7> return
```

For additional information on configuring RADIUS groups and servers, refer to the *Net-Net OS-E – Objects and Properties Reference*.

Including the RADIUS group

When you configure RADIUS groups, you include one or more groups with the VSP RADIUS accounting configuration. This tells the VSP what RADIUS servers to use when forwarding RADIUS accounting requests.

CLI session

The following CLI session includes the RADIUS groups named **aaaGroup1** and **aaaGroup2** with the VSP RADIUS accounting configuration.

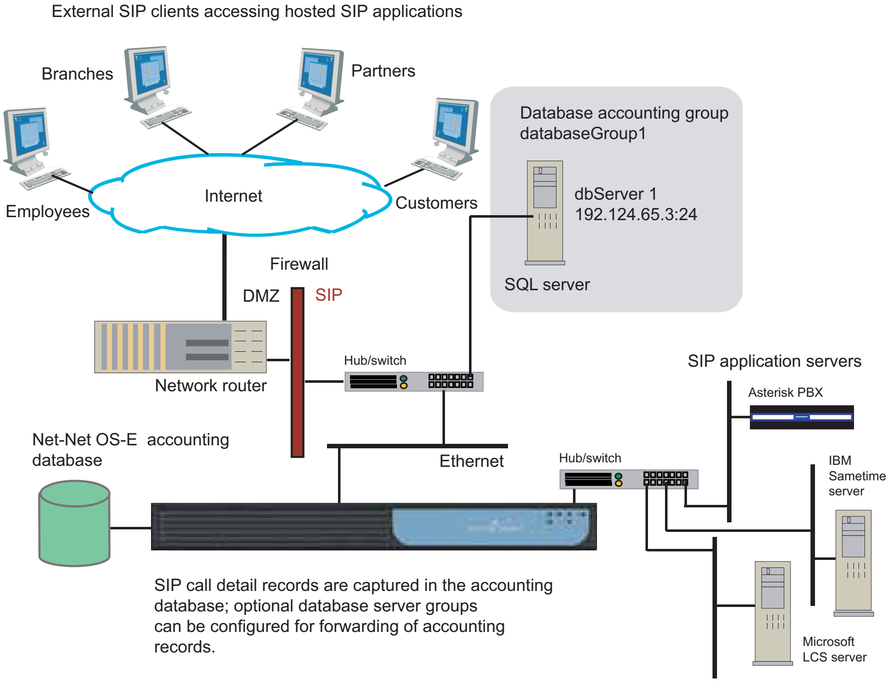
```
NNOS-E> config vsp
config vsp> config accounting
config accounting> config radius
config radius> set admin enabled
config radius> set group vsp radius-group aaaGroup1
config radius> set group vsp radius-group aaaGroup2
config radius> show

vsp
 accounting
  radius
   admin enabled
   group vsp\radius-group aaaGroup1
   group vsp\radius-group aaaGroup2
```

When using the set group command, specify the CLI path where you created the Radius group.

Configuring the accounting database

The AA-SBC accounting database is a subsystem that captures and stores SIP call detail records. If configured, these records can be forwarded to remote SQL database servers such as Oracle and Postgres where the call detail records are used with other accounting and billing applications. Access to a remote database group and server is restricted by configured user names and passwords.

Accounting policies direct SIP call detail records to specific accounting groups and servers. If you do not configure one or more remote database groups and servers, the SIP call detail records are stored in the AA-SBC accounting database only. The following image illustrates a sample network with a database server group.


External SIP clients accessing hosted SIP applications
Branches
Partners
Employees
Internet
Customers
Database accounting group
databaseGroup1
dbServer 1
192.124.65.3:24
SQL server
Firewall
DMZ
SIP
Network router
Hub/switch
SIP application servers
Asterisk PBX
Net-Net OS-E accounting database
Ethernet
Hub/switch
IBM Sametime server
Microsoft LCS server
SIP call detail records are captured in the accounting database; optional database server groups can be configured for forwarding of accounting records.

CLI session

The following CLI session creates the accounting database group named **databaseGroup1,** creates the associated server named **dbServer1**, and sets the group and server operating properties.

```
NNOS-E> config vsp
```

```
config vsp> config accounting
config accounting> config database
config accounting> set admin enabled
config database> config group databaseGroup1
Creating ‘group databaseGroup1’
config group databaseGroup1> set admin enabled
config group databaseGroup1> set mode duplicate

config group databaseGroup1> config server dbServer1
Creating ‘server dbServer1’
config group databaseGroup1> set admin enabled
config group databaseGroup1> set type sqlserver 192.124.65.3 24 srvr1
config group databaseGroup1> set username frank
config group databaseGroup1> set password-tag kj3k2
```

In this session, the duplicate mode algorithm issues a duplicate accounting request to all servers in the accounting group. A duplicate accounting request uses the same client source IP address and source UDP port. If you configure multiple database servers in the database group, the fail-over algorithm forwards one accounting request to each secondary servers should the current session fail.

The **databaseGroup1** accounting group is a default group for all non-policy governed accounting database requests between the AA-SBC platform and the database servers.

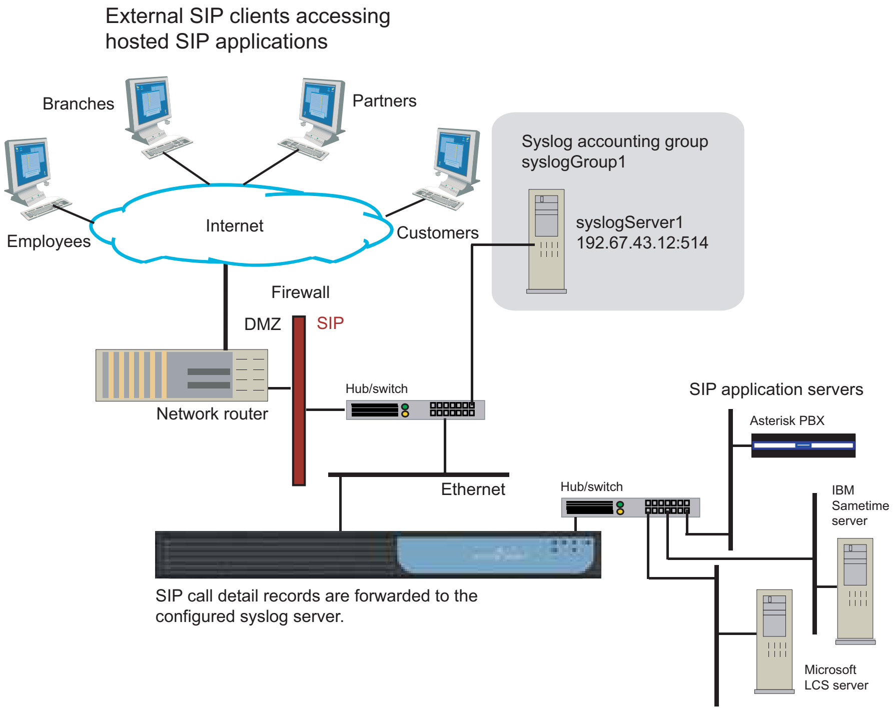
Note: If you set the server **type** to *local* while using the local database as the accounting target, set the **username** and the **password-tag** to *postgres*. If you edit the **username** and **password-tag** properties to anything other than *postgres*, data will not be written to the database.

For additional information on configuring accounting database groups and servers, refer to the *Net-Net OS-E – Objects and Properties Reference*.

Configuring syslog

Syslog allows you to log accounting information to a remote server using the configured syslog format: Acme Packet, CSV, tabular, or XML format. When enabled, SIP call detail records are forwarded to the specified syslog accounting group and server. The following image illustrates a sample network.

Figure 0-1. Sample AA-SBC network with syslog server configuration



CLI session

The following CLI session creates the syslog accounting group named **syslogGroup1**, specifies the associated syslog server at **192.167.43.12** on port **514**, and sets the syslog group and server operating properties.

```
NNOS-E> config vsp
config vsp> config accounting
config accounting> config syslog
config syslog> set admin enabled
config syslog> config group syslogGroup1
Creating ‘group syslogGroup1’
config group syslogGroup1> set admin enabled
config group syslogGroup1> set format csv

config group syslogGroup1> config server 192.167.43.12:514
Creating ‘server 192.167.43.12:514’
config server 192.167.43.12:514> set admin enabled
config server 192.167.43.12:514> set name syslogserver1
config server 192.167.43.12:514> set facility local0
config server 192.167.43.12:514> set priority info
config server 192.167.43.12:514> set include-timestamp true
```

In this session, **syslogGroup1** uses Comma-Separated Values (CSV) 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. The **syslogGroup1** accounting group is a default group for all non-policy governed accounting database requests between AA-SBC and the syslog servers.

The syslog server at IP address and port **192.67.43.12:514** is enabled with the operator-defined name **syslogserver1**. The facility (local0 to local7) specifies where SIP call detail records are logged. Syslog facilities help isolate the origin of messages written to the syslog server. The syslog priority (info, emergency, alert, etc.) sets the message priority to be associated SIP call detail records. All AA-SBC accounting and SIP call detail records are assigned this priority before they are forwarded to the syslog server. A time stamp can also be applied to each accounting record.

For additional information on configuring accounting database groups and servers, refer to the *Net-Net OS-E – Objects and Properties Reference*.

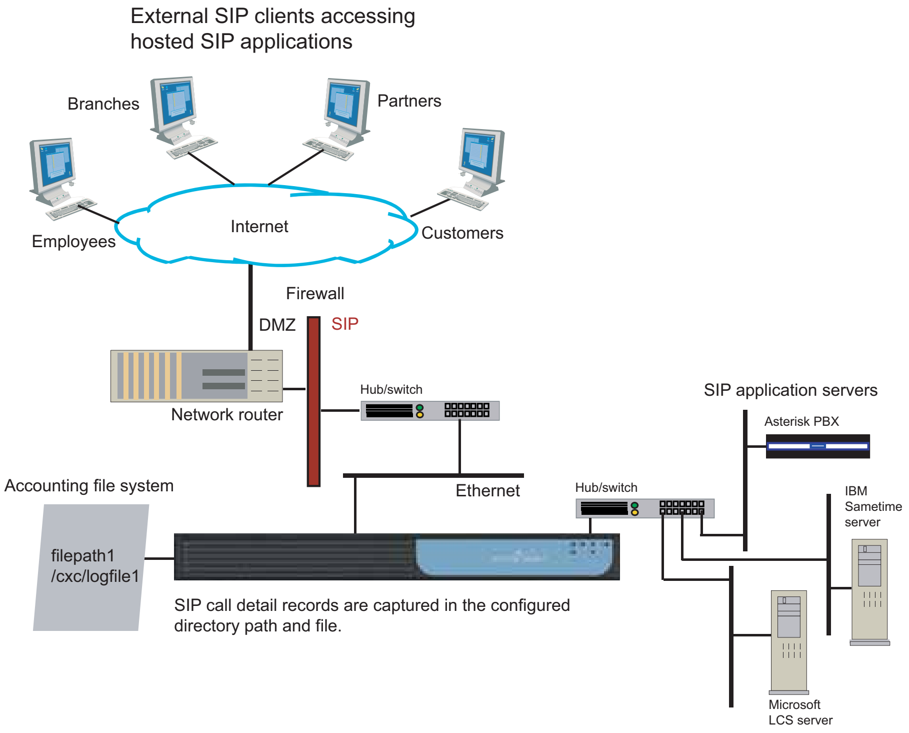
Configuring the file system

The accounting file system allows you to direct SIP call detail records to a named directory path and file using a specified format: CSV, tabular., Acme Packet text file format, or to a temporary output file in the case of postgres format.

There are two states that the file system cycles through as it processes raw CDRs and writes to the output file.

- Clear—The target is ready to write.
- Writing—The target is writing to the output file.

The following image illustrates a sample network.



CLI session

The following CLI session creates the file system group named **filePath1,** specifies the format, file path, and target file name, and sets the file system operational properties.

```
NNOS-E> config vsp
config vsp> config accounting
config accounting> config file-system
config file-system> set admin enabled
config file-system> config path filePath1
Creating ‘path filePath1’
config path> set admin enabled
config path> set format csv
config path> set call-field-filter recorded
config path> set file-path \cxc\logfile1.csv
config path> set roll-over never
config path> set purge-old-logs true
config path> set retention-period 1 days
```

In this session, **filePath1** uses Comma-Separated Values (CSV) 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. The AA-SBC target file path is **\cxc\logfile1.csv**, where **logfile1.csv** is the name of the file to which SIP call detail records are forwarded.

The **roll-over** property maintains and keeps the original time as it was first applied to the log file. The log file will continue to build under this time stamp. The **filePath1** file system accounting group is a default target group for capturing all non-policy governed SIP call detail records.

For additional information on configuring accounting database groups and servers, refer to the *Net-Net OS-E – Objects and Properties Reference*.

Configuring an External File System Target

The external-file-system target allows you to send accounting records from the AA-SBC Management System to a remote system. The target is able to read raw CDRs and write this information to a temporary output file in the format you specify during configuration.

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.

If the configuration is modified or deleted, any files currently being processed are sent immediately and without retries. If the target is in the blocked state, there are two files immediately sent and if the target is in the sending or writing states, one file is sent. The modification or deleted is applied only after the send completes, successfully or not.

If there is a failure when sending a file to the external target, the send is retried every 30 seconds for an hour. After an hour, the send is retried once every hour until it succeeds.

The following is the format of the output file:

```
<target-name>-<yyyy-mm-dd-hh-mm>-<processingtype>-<seq-no>.<xtn>
```

- target-name—Name specified in the configuration.
- yyyy-mm-dd-hh-mm—The timestamp when the output file is created.
- processingtype—Hourly, daily, never.
- xtn—.csv, .tab, .cov, or .pg

CLI Session

The following CLI session creates the external file system target, sets the target format, URL address, and CDR processing.

```
NNOS-E>config vsp
config vsp>config accounting
config accounting>config external-file-system
config external-file-system>config url 7
Creating 'url test'
config url 7>
```

```
   config url test>set admin enabled
   config url test>set format csv
   config url test>set url ftp://lalenchery:BillGates#1@10.33.5.10:/
acct/test/
   config url test>set cdr-processing batch 10
   config url test>
```

For additional information on configuring external file system targets, refer to the NNOS-E Objects and Properties Reference.

Configuring Diameter

The Diameter protocol, as described in RFC 3588, provides Authentication, Authorization and Accounting (AAA) services for applications such as IP mobility and SIP multimedia communications sessions. A AA-SBC platform (SIP proxy), operating as 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.

Diameter is also an essential component for the least cost routing (LCR) functionality. Refer to the *Net-Net OS-E – Session Services Configuration Guide* for detailed information on LCR.

Creating the Diameter accounting group

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 VSP configurations allows the AA-SBC platform (the Diameter client) to perform user authentication and forward SIP call detail records to Diameter servers.This means that you have flexibility to create as many unique Diameter groups as you need, and include them with the VSPs of your choice.

CLI session

The following CLI session creates the Diameter accounting group named **diameterGroup1** and sets the group operational properties.

```
NNOS-E> config vsp
config vsp> config diameter-group 1
Creating ‘diameter-group 1’
```

```
config diameterGroup1> set admin enabled
config diameterGroup1> set authentication-mode round-robin
config diameterGroup1> set application sip
config diameterGroup1> set origin-host text
config diameterGroup1> set origin-realm text
config diameterGroup1> set default-destination-realm text
```

In this session, the **authentication-mode**, sets the Diameter group authentication operational algorithm. This example allows continued authentication requests to primary and secondary servers until a valid authentication response is received (round-robin).

The **application** setting specifies the target application for the servers in this Diameter group. Choose **SIP** for standard AAA activities, **3GPPRx** for inter-operation with the Camiant policy server (enabled with the Rx object), and **Routing** for least-cost-routing between clusters.

The **origin-host** specifies the text written to the Origin-Host attribute field in any Diameter *requests* it sends. This should be the AA-SBC domain name.

The **origin-realm** specifies the text written to the Origin-Realm attribute field in any Diameter requests it sends. This should be the AA-SBC domain name.

The **default-destination-realm** specifies the text written to the Destination-Realm attribute field in any Diameter responses it sends. This setting operates with the 3Gpp Rx application.

Configuring Diameter servers

You can configure multiple Diameter servers in the Diameter group, and you identify each server using a unique name, authentication port, and other operational settings.

CLI session

The following CLI session creates two numbered Diameter servers and sets the operational properties for Diameter requests and responses between the AA-SBC platform and the Diameter peers.

```
NNOS-E> config vsp
config vsp> config diameter-group 1
Creating ‘diameter-group 1’
config diameterGroup1> set admin enabled
config group diameterGroup1> config server diameterServer1
Creating ‘server diameterServer1>
```

```
config diameterServer 1> set admin enabled
config diameterServer 1> set port 3868
config diameterServer 1> set transport tcp
config diameterServer 1> set authentication-port 3868
config diameterServer 1> set request-timeout 2
config diameterServer 1> set window 8
config diameterServer 1> set priority 1

NNOS-E> config vsp
config vsp> config diameter-group 1
Creating ‘diameter-group 1’
config diameterGroup1> set admin enabled
config group diameterGroup1> config server diameterServer2
Creating ‘server diameterServer2>
config diameterServer 2> set admin enabled
config diameterServer 2> set port 3868
config diameterServer 2> set transport tcp
config diameterServer 2> set authentication-port 3868
config diameterServer 2> set request-timeout 2
config diameterServer 2> set window 8
config diameterServer 2> set priority 1
```

For additional information on configuring Diameter groups and servers, refer to the *Net-Net OS-E – Objects and Properties Reference*.

Configuring Diameter interfaces and ports

The **diameter** configuration object under the **box\interface\ip object** identifies the IP interface on which the Diameter server application resides. This is the AA-SBC interface that listens for incoming Diameter connections. This interface must be configured on each AA-SBC domain that is referenced by a server in a Diameter group.

CLI session

```
config box> config interface eth3
config interface eth3> config ip A

config ip A> config diameter
config diameter> set admin enabled
config diameter> set origin-host text
config diameter> set origin-realm text

config diameter> config port 3868
Creating ‘port 3868’
config port 3868> set admin enabled
config port 3868> set transport tcp
config port 3868> set application sip
config port 3868> set peer-access-control transport
config port 3868> set peer ipaddress
```

The **origin-host** setting specifies the text written to the Origin-Host attribute field in any Diameter *responses* it sends. This should be the DNS name of the AA-SBC domain you are configuring.

The **origin-realm** specifies the text written to the Origin-Realm attribute field in any Diameter *responses* it sends. This should be the AA-SBC domain name.

The **port** configuration specifies properties for incoming Diameter connections. The **application** setting sets the application that the incoming connection must be running to use this port. Choose **SIP** for standard AAA activities, **3GPPRx** for inter-operation with the Camiant policy server (enabled with the Rx object), and **Routing** for least-cost-routing between clusters.

The **peer-access-control** setting specifies how AA-SBC controls incoming peer connections. You can select to allow incoming connection from all peers or from peers on a configured list based on address or Host-IP-Address AVP.

The **peer** setting 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.** Indicate the peer by specifying the peer IP address.

Configuring archiving

The **accounting/archiving** object allows you to configure an archiving location for SIP call detail records. 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).

You must configure an archiving server in one of the archiving sub-objects for the archiving mechanism to work:

- **windows-share** — Archiving of accounting and SIP call records to a selected Windows server partition.
- **ftp-server** — Archiving of accounting and SIP call records to a selected FTP server.
- **http-server** — Archiving of accounting and SIP call records to a selected HTTP server

- **smtp-server** — Enables archiving of accounting and SIP call records to a selected Simple Mail Transfer Protocol (SMTP) server. When enabled, the AA-SBC sends out the archives in the form of an email attachment to the specified destination mailbox.
- **db-server** — Archiving of accounting and SIP call records to a selected database server.
- **local** — Archiving of accounting and SIP call records to a location on the AA-SBC platform.

The following CLI session configures a remote database server for archiving of SIP call detail records.

CLI session

```
NNOS-E> config vsp
config vsp> config accounting
config accounting> config archiving
config archiving> config db-server database1
Creating ‘db-server database1’
config db-server database1> set admin enabled
config db-server database1> set username admin
config db-server database1> set password-tag xyz123abc
config db-server database1> set server 192.168.10.10
config db-server database1> set url www.companyABC.com
config db-server database1> set driver-class com.oracle.jdbc.Driver
```

If you are archiving using the **http-server** method, a server-side script designed to be run with Apache 2.0 and perl 5.8.5 on Linux is needed to handle the POST requests that are sent from the AA-SBC to transfer the archive zip files to the server. The following is an example:

```
#!/usr/bin/perl
#---Modify the above line to match the location of perl on your system---

#---This script has been tested running with AA-SBC software version 3.5.2
    sending
#to Apache 2.0.52 running on Redhat EL4 Linux with perl 5.8.5---

#---Make sure to modify file permissions for this script so that it can
#be executed by the user running the httpd daemon.---

#---Note this script is provided as an example, which makes no attempt to
    validate
#the values pulled from the HTTP POST to ensure execution security---

#---Require strict syntax---
use strict;
use warnings;
```

```
#---Use the CGI library provided with perl - CGI.pm---
use CGI;
#---The below lines are an example of code, provided as-is, used to take
#the multipart/form-data from an HTTP POST to this script, which
#apache presents on STDIN and write it out to the disk in the
#directory specified in the variable above, using the same filename
#presented in the HTTP POST---

#---Instantiate CGI object---
my $cgi = new CGI;
my %params = $cgi->Vars;

#---Get proper filehandle from unknown file param name---
my $filehandle;
my $anon_param;
foreach my $param (keys %params) {
$anon_param = "$params{$param}" if (("$param" ne "name") && ("$param" ne
    "path"))
};

$filehandle = $cgi->param($anon_param);

#---Pull target directory from "path" cgi variable; this comes from the
    "directory"
#in the AA-SBC config. Note: leave off the trailing slash------

#---Make sure to modify file permissions for target directory so that it can
#be executed and written to by the user running the httpd daemon.---
my $dir = $cgi->param('path');
#---Pull target filename from "name" cgi variable
#---Assemble directory and filename---
my $name = $cgi->param('name');
my $fullname = "$dir/$name";

#---Write out the file from the HTTP POST---
open(LOCAL, ">$fullname") or die $!;
binmode LOCAL;
while(<$filehandle>) { print LOCAL $_; }
close(LOCAL);

#---Needed for 200OK response---
print $cgi->header( "text/plain" ), "File received.";
```

The following example displays the way the AA-SBC must be configured for the http-server archiving to work:

```
    config archiving> config http-server server1
    config http-server server1> set admin enabled
    config http-server server1> set directory /tmp/archives
    config http-server server1> set url http://10.0.0.1/cgi-bin/
archive_http_upload_example.pl
    config http-server server1> set timeout 60000
```

- The server needs to be configured to allow CGI scripts.
- The script needs to be placed in the “cgi-bin” directory and given execute permission for the user running the server.
- The URL needs to include the name of the script.

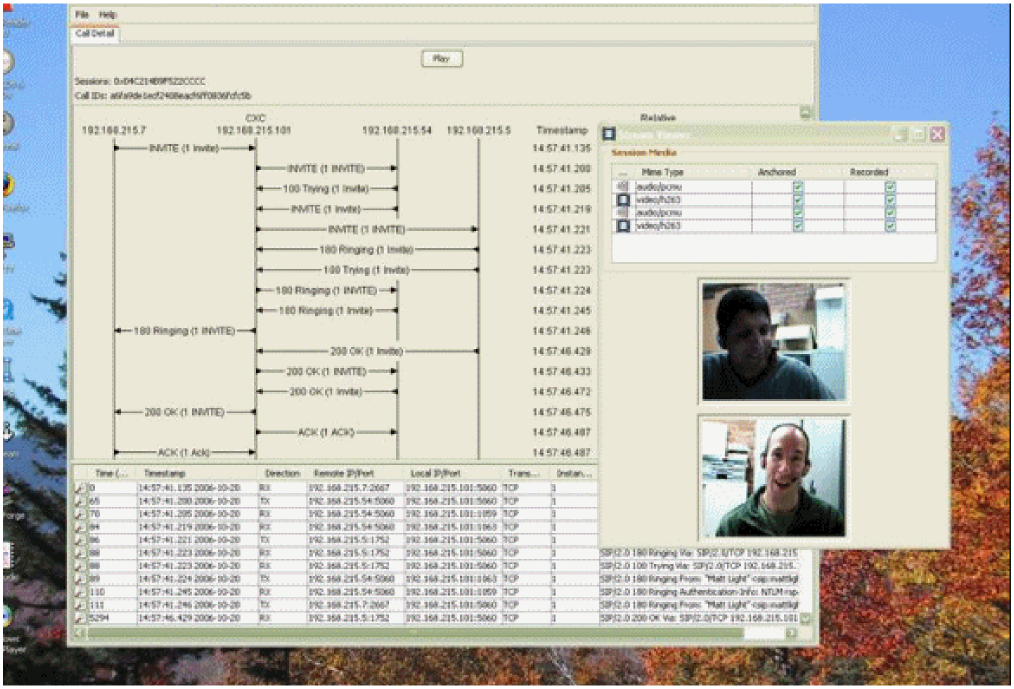
- The directory needs to have "write" permissions for the user running Apache. This argument gets passed through the HTTP POST to the scripted. It is used to determine to which directory on the server the archive file is written.

For additional information on archiving accounting records, refer to the *Net-Net OS-E – Objects and Properties Reference.*

Using the AA-SBC archive viewer

The archive viewer is a standalone utility that displays information and plays video recordings from archive files that have been stored locally on a client PC. The viewer allows you to see the call diagram and message details without having to run the AA-SBC Management System.

The following image illustrates a sample Archive Viewer display.



Sessions: 0x04C214B9F522CCCC
Call IDs: a6fa9de1ecf2408eacf6f0806fcfc5b

192.168.215.7	CXC 192.168.215.101	192.168.215.54	192.168.215.5	Timestamp
INVITE (1 Invite) →				14:57:41.135
	INVITE (1 INVITE) →			14:57:41.200
	← 100 Trying (1 Invite)			14:57:41.205
	← INVITE (1 Invite)			14:57:41.219
	INVITE (1 INVITE) →			14:57:41.221
	← 180 Ringing (1 Invite)			14:57:41.223
	← 100 Trying (1 Invite)			14:57:41.223
	180 Ringing (1 INVITE) →			14:57:41.224
	← 180 Ringing (1 Invite)			14:57:41.245
← 180 Ringing (1 INVITE)				14:57:41.246
	← 200 OK (1 Invite)			14:57:46.429
	200 OK (1 INVITE) →			14:57:46.433
	← 200 OK (1 Invite)			14:57:46.472
← 200 OK (1 INVITE)				14:57:46.475
	ACK (1 ACK) →			14:57:46.487
ACK (1 Ack) →				14:57:46.487

Mime Type	Anchored	Recorded
audio/pcmu	☑	☑
video/h263	☑	☑
audio/pcmu	☑	☑
video/h263	☑	☑

Time (...	Timestamp	Direction	Remote IP/Port	Local IP/Port	Trans...	Instan...	
0	14:57:41.135 2006-10-20	RX	192.168.215.7:2667	192.168.215.101:5060	TCP	1	
65	14:57:41.200 2006-10-20	TX	192.168.215.54:5060	192.168.215.101:5060	TCP	1	
70	14:57:41.205 2006-10-20	RX	192.168.215.54:5060	192.168.215.101:1059	TCP	1	
84	14:57:41.219 2006-10-20	RX	192.168.215.54:5060	192.168.215.101:1063	TCP	1	
86	14:57:41.221 2006-10-20	TX	192.168.215.5:1752	192.168.215.101:5060	TCP	1	
88	14:57:41.223 2006-10-20	RX	192.168.215.5:1752	192.168.215.101:5060	TCP	1	SIP/2.0 180 Ringing Via: SIP/2.0/TCP 192.168.215
88	14:57:41.223 2006-10-20	RX	192.168.215.5:1752	192.168.215.101:5060	TCP	1	SIP/2.0 100 Trying Via: SIP/2.0/TCP 192.168.215.
89	14:57:41.224 2006-10-20	TX	192.168.215.54:5060	192.168.215.101:1063	TCP	1	SIP/2.0 180 Ringing From: "Matt Light" <sip:mattlig
110	14:57:41.245 2006-10-20	RX	192.168.215.54:5060	192.168.215.101:1059	TCP	1	SIP/2.0 180 Ringing Authentication-Info: NTLM rsp-
111	14:57:41.246 2006-10-20	TX	192.168.215.7:2667	192.168.215.101:5060	TCP	1	SIP/2.0 180 Ringing From: "Matt Light" <sip:mattlig
5294	14:57:46.429 2006-10-20	RX	192.168.215.5:1752	192.168.215.101:5060	TCP	1	SIP/2.0 200 OK Via: SIP/2.0/TCP 192.168.215.101

The Archive Viewer is contained in a ZIP file included with the AA-SBC release software.

Perform the following steps on a Windows PC, which is the only supported platform for the Archive Viewer:

1. Using your web browser, go to your customer support site and log on.
2. Download the appropriate image file to a location on your PC.
3. Double-click the .ZIP file, then select **Extract All**. A separate folder will be created using the same name, minus the .ZIP extension.
4. Open the folder that you just created, then double-click the Archive Viewer file.
5. Select **File->Open Archive**, or **File->Stream Viewer** to browse for the archived file. The Stream Viewer replays and mixes the two audio streams (one in each direction) with the video streams (one in each direction).

Note: You must configure AA-SBC with both accounting and media recording enabled. You can enable archiving to periodically send the recorded files to a workstation, or you can create individual session archives on demand from the AA-SBC Management System **Call Logs** screen.

5. Configuring Domain Name Systems (DNS)

About this chapter

The chapter covers DNS configurations on the AA-SBC platform.

Domain Name System (DNS) overview

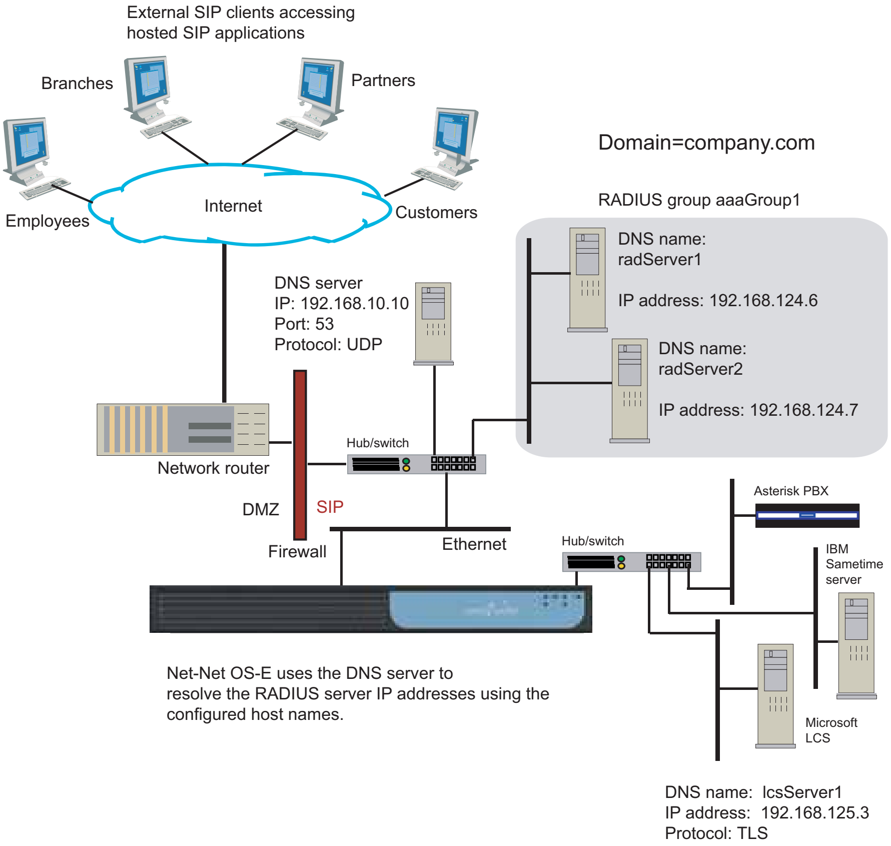
Domain Name System (DNS) servers are responsible for translating Internet domain and host names to IP addresses. DNS converts the name entered on a Web browser address bar to the IP address of the Web server that hosts that particular Web site. DNS uses a distributed database to store this name and address 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's IP address. If the DNS server cannot service the request, it forwards the request to other DNS servers until the IP address is resolved, completing the Internet client request.

AA-SBC maintains a cache of query responses—positive responses were successful and negative (reject) responses failed. This response is the DNS resource record, allowing AA-SBC to consult its cache for mapping information before querying a server.

RADIUS and Diameter group accounting configurations, for example, require that you configure DNS to resolve the IP addresses associated with RADIUS and Diameter servers being used to capture call detail records.

The following image illustrates a sample network with a DNS server that resolves RADIUS server IP addresses using the domain name.



Net-Net OS-E uses the DNS server to resolve the RADIUS server IP addresses using the configured host names.

Configuring the DNS resolver

AA-SBC functions as a DNS client (resolver) that forwards requests for IP address resolutions, but does not act as a server in accepting requests. As a resolver, AA-SBC obtains resource records from DNS servers on behalf of resident or requesting applications. You must configure the resolver function before other objects within the DNS configuration object.

Note: You must configure the settings of the **resolver** object before setting other objects under DNS.

The DNS object configures the AA-SBC domain name, one or more DNS servers, and static mapping between host names and addresses. You can also configure static service locations, naming authority pointers, and how to resolve negative entries.

CLI session

The following CLI session configures and enables the DNS resolver, sets the domain name to be used for DNS mappings, sets the DNS server IP address, port number and transport protocol, and the DNS query properties.

```
NNOS-E> config vsp
config vsp> config dns
config dns> config resolver
config resolver> set admin enabled
config resolver> set server 192.168.10.10 UDP 54
config resolver> set query-timeout 10
config resolver> set query-retries 5
config resolver> set cache-poll-interval 60
```

The **query-timeout** property specifies the time, in seconds (between 1 to 10), that a DNS lookup can go unanswered before it times out. The **query-retries property** specifies the number of DNS query (lookup) retries to execute if a DNS query times out. Enter a number of retries between 0 to 5, where 0 indicates no retries.

The **cache-poll-interva**l property specifies the number of seconds that AA-SBC waits between refreshing the cache. The interval controls the rate at which AA-SBC polls the location-cache to purge stale location bindings.

Configure as many DNS servers as you need. Refer to the *Net-Net OS-E – Objects and Properties Reference* for information on additional settings.

Configuring DNS hosts and IPs

For each host in your network domain, you need to statically map IP addresses to host names. The **host** object requires that you supply a ***name*** variable. This is the name of an Internet node, for example, a SIP server, a RADIUS server, or a PC in your network.

You can enter:

- An existing name and new address — the corresponding address is mapped to the name for use in DNS lookups, or

- A new name and existing address — the system creates a named entry for DNS use.

CLI session

The following DNS session configures the DNS host name for the RADIUS server named **radServer1** and sets the IP address to be returned in DNS lookups.

```
NNOS-E> config vsp
config vsp> config dns
config dns> config host radServer1
Creating ‘host radServer1’
config host radServer1> set address 192.168.124.6
```

The following DNS session configures the DNS host name for the SIP server named **lcsServer1** and sets the IP address to be returned in DNS lookups.

```
NNOS-E> config vsp
config vsp> config dns
config dns> config host lcsServer1
Creating ‘host lcsServer1
config host lcsServer1> set address 192.168.125.3
```

Mapping SIP services

The DNS **service** object allows you to statically map SIP services to specific SIP servers. Using a configured rule, DNS resolves the SIP service and maps the service to a specific SIP server. By adding DNS server resource (SRV) records for each SIP service, SRV records provide contacts for the specific DNS servers.

The **rule** property establishes the preference level for selecting a named SIP service if you configure multiple SIP service mappings. Configuring the **service** object for each SIP service establishes the sequence to use when contacting the configured SIP servers.

CLI session

The following CLI session maps the TLS service on the **company.com** domain. DNS resolves the TLS service to **lcsServer1** using the configured rule (port, priority, and weight settings).

```
NNOS-E> config vsp
config vsp> config dns
config dns> config service company.com tls
```

```
Creating ‘service company.com tls’
config service company.com> set rule lcsServer1.company.com 5001 10 5
```

Configuring NAPTR

The Naming-authority pointer (called NAPTR) creates a static mapping of service information to a specific server or domain name. This mapping performs DNS lookups for requests in cases where the AA-SBC platform cannot determine either the protocol or port of the destination.

Naming-authority pointer (NAPTR) records contain rules for converting each request to the correct configured service. Because each transport service over SIP is viewed as a different service (TCP, UDP, or TLS), 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 the protocol and the port for the destination.

CLI session

The following CLI session sets the NAPTR rules (protocol, order, preference) for SIP TLS, TCP and UDP services on the **company.com** domain. DNS uses the configured SIP services (TLS, TCP, UDP) to resolve the destination SIP server, using exact matching of the **company.com** domain name.

```
NNOS-E> config vsp
config vsp> config dns
config dns> config naptr company.com
Creating ‘naptr company.com’
config naptr company.com> set match exact
config naptr company.com> set rule TLS 1 10
config naptr company.com> set rule TCP 2 10
config naptr company.com> set rule UDP 3 10
```

For more information on NAPTR and DNS on the AA-SBC platform, refer to the *Net-Net OS-E – Objects and Properties Reference*

Configuring DNS rejections

You can instruct DNS to ignore lookups that involve certain domain names. The DNS **reject** object requires that you supply a host name, service name, domain name, or IP address. Any request containing the specified name will be rejected.

Set the **type** property to identify which record type you are entering:

- A — IPv4 address
- AAAA — IPv6 address
- PTR — Address to name mapping
- NAPTR — NAPR rule

CLI session

The following CLI session rejects DNS lookups that involve the domain named **`evilBadGuy.com`**., using the IPv4 address, matching the exact domain name as entered.

```
NNOS-E> config vsp
config vsp> config dns
config dns> config reject badNetwork.com naptr
Creating ‘reject badNetwork.com naptr’
config reject badNetwork.com> set match exact
```

For more information on DNS rejections on the AA-SBC platform, refer to the *Net-Net OS-E – Objects and Properties Reference*.

6. Managing certificates and keys with LCS and OCS

About this chapter

The chapter covers certificate and key management on the AA-SBC platform. Certificates and keys are required for SIP sessions over TLS transports between the AA-SBC platform and a Microsoft LCS/OCS server

Transport Layer Security overview

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 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 carrier and enterprise SIP applications.

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)

Acquiring certificates

The certificates required for the TLS protocol and SIP session establishment must come from a valid certificate authority (CA).

A CA is a company that issues a certificate stating that the CA guarantees that the public key contained in the certificate belongs to the person or organization noted in the certificate. The CA verifies the identify of the applicant's so that users can trust certificates issued by that CA to belong to the people and data identified in it, and not to an imposter.

The certificate file can be in the following formats:

- PKCS#12 — Public Key Cryptography Standard #12 format from Microsoft IIS Version 5 (binary)
- PEM — Privacy-enhanced mail (PEM) encoded format from any OpenSSL-based Web server (ASCII)

If you do not have a certificate file for your network, you need to contact a valid CA who will issue (for a fee) a certificate in PKCS #12 or PEM format.

Installing the CA chain file on LCS/OCS servers

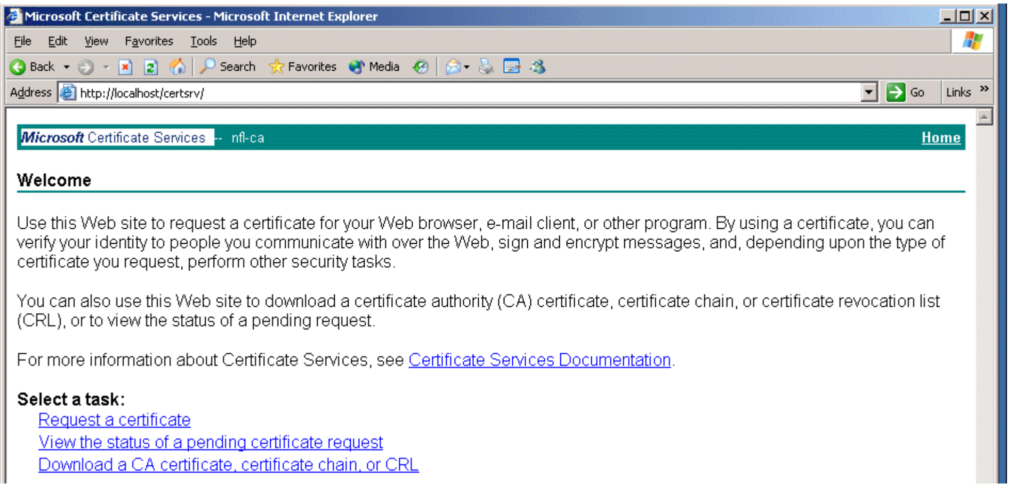
The section covers the steps for installing the CA chain file on an LCS/OCS server.

1. From the LCS/OCS Server, go to the HTTP location where you are running the Microsoft Certificate Authority. For example:

   http://localhost/certserv

   The Microsoft Certificate Services window appears.

2. If you have not already done so, log on as **Administrator** of the CA.



http://localhost/certsrv/

Microsoft Certificate Services -- nfl-ca **Home**

Welcome

Use this Web site to request a certificate for your Web browser, e-mail client, or other program. By using a certificate, you can verify your identity to people you communicate with over the Web, sign and encrypt messages, and, depending upon the type of certificate you request, perform other security tasks.

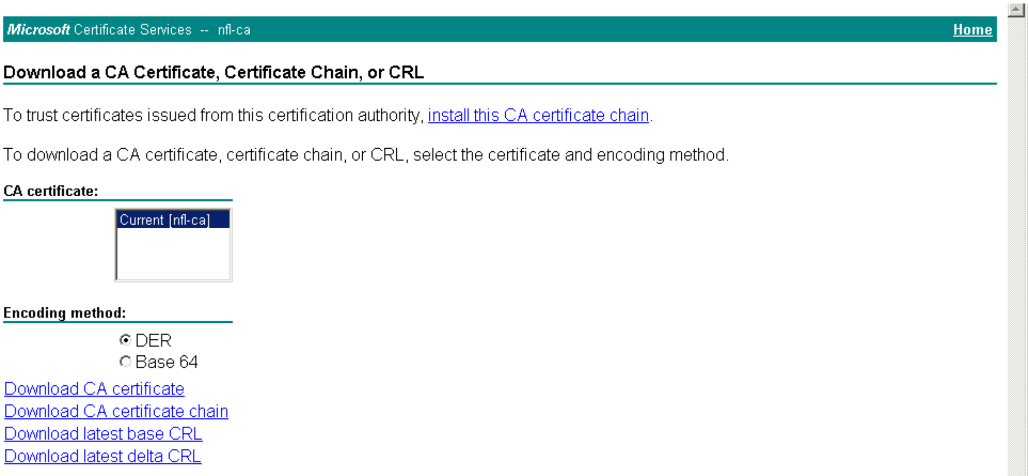
You can also use this Web site to download a certificate authority (CA) certificate, certificate chain, or certificate revocation list (CRL), or to view the status of a pending request.

For more information about Certificate Services, see Certificate Services Documentation.

Select a task:

Request a certificate
View the status of a pending certificate request
Download a CA certificate, certificate chain, or CRL

3. Select **Download a CA certificate, certificate chain, or CRL** to display the next screen.



Microsoft Certificate Services -- nfl-ca

Download a CA Certificate, Certificate Chain, or CRL

To trust certificates issued from this certification authority, install this CA certificate chain.

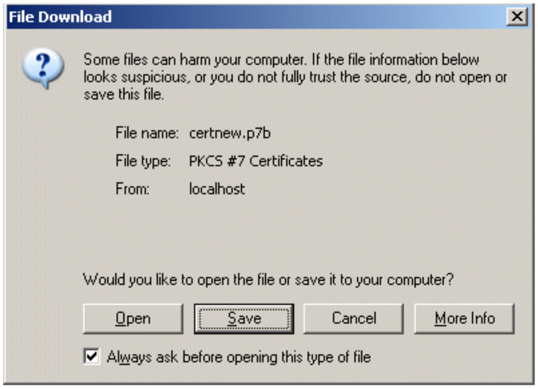
To download a CA certificate, certificate chain, or CRL, select the certificate and encoding method.

CA certificate: Current [nfl-ca]

Encoding method:

● DER
○ Base 64

4. Select **Download a certificate chain** to display the File Download window.



File Download

Some files can harm your computer. If the file information below looks suspicious, or you do not fully trust the source, do not open or save this file.

File name: certnew.p7b

File type: PKCS #7 Certificates

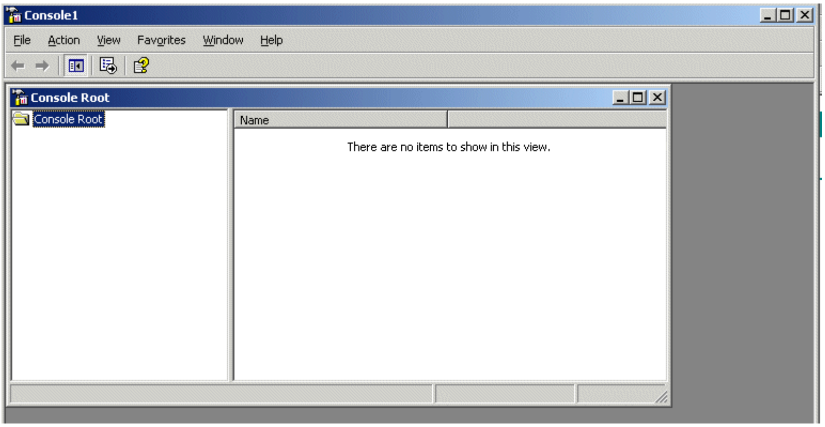
From: localhost

Would you like to open the file or save it to your computer?

☑ Always ask before opening this type of file

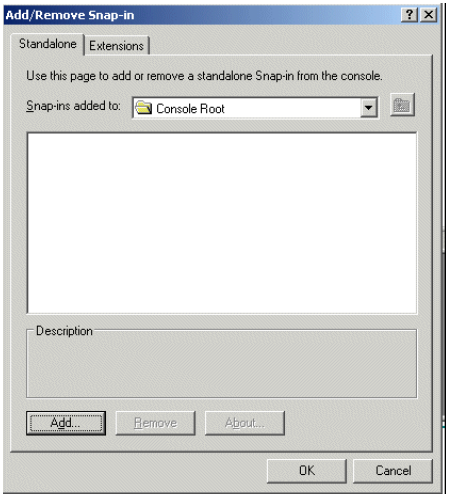
5. Click **Save** to save the file to a location on the LCS/OCS server. Click **Close** when the Download Complete screen appears.

6. At the **Start** button, click **Run**, then enter **mmc** on the command line. Click OK to display the Console Root window.



There are no items to show in this view.

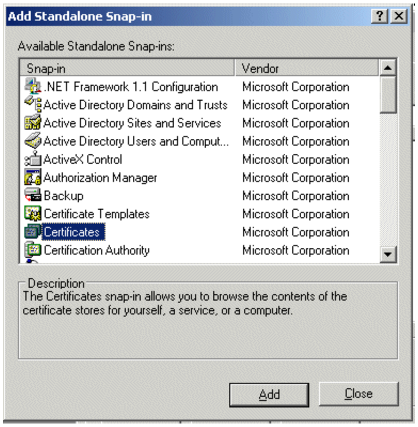
7. From the Console Root window, go to the pull-down menu and select **File ->Add/ Remove Snap-in.** The Add/Remove Snap-in window appears.



Use this page to add or remove a standalone Snap-in from the console.

Snap-ins added to: Console Root

8. From the Add/Remove Snap-in window, click **Add**, then select **Certificates** from the list of Snap-in options in the Add Standalone Snap-in window.



Available Standalone Snap-ins:

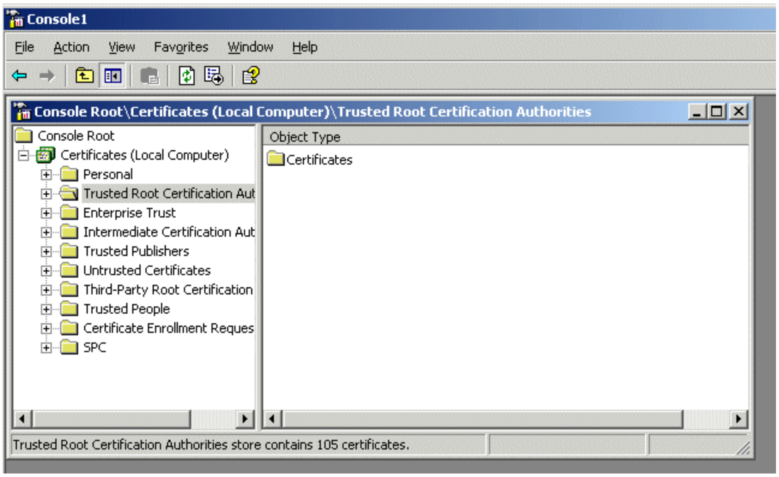
Snap-in	Vendor
.NET Framework 1.1 Configuration	Microsoft Corporation
Active Directory Domains and Trusts	Microsoft Corporation
Active Directory Sites and Services	Microsoft Corporation
Active Directory Users and Comput...	Microsoft Corporation
ActiveX Control	Microsoft Corporation
Authorization Manager	Microsoft Corporation
Backup	Microsoft Corporation
Certificate Templates	Microsoft Corporation
Certificates	Microsoft Corporation
Certification Authority	Microsoft Corporation

Description
The Certificates snap-in allows you to browse the contents of the certificate stores for yourself, a service, or a computer.

9. Click **Add->Computer Account->Next->Local computer.** Click **Finish**.

10. Click **Close,** then **OK**, to return to the Console Root window.

11. From the Console Root window, select **Certificates (Local computer)->Trusted Root Certification Authorities->Certificates**.


Console1
File
Action
View
Favorites
Window
Help
Console Root\Certificates (Local Computer)\Trusted Root Certification Authorities

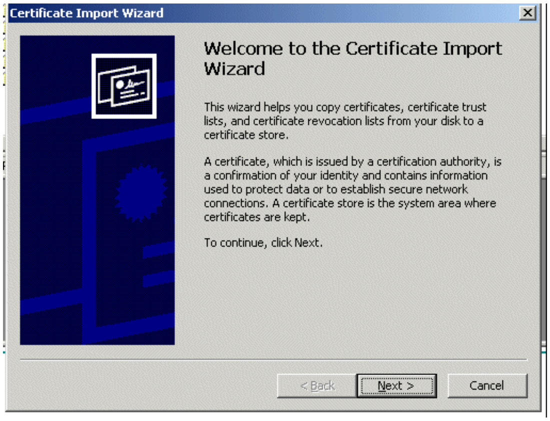
- Console Root
  - Certificates (Local Computer)
    - Personal
    - Trusted Root Certification Aut
    - Enterprise Trust
    - Intermediate Certification Aut
    - Trusted Publishers
    - Untrusted Certificates
    - Third-Party Root Certification
    - Trusted People
    - Certificate Enrollment Reques
    - SPC

Object Type
Certificates

Trusted Root Certification Authorities store contains 105 certificates.

12. Right click **Certificates->All Tasks->Import**.

13. Import the CA chain file using the Certificate Import Wizard.



Certificate Import Wizard

Welcome to the Certificate Import Wizard

This wizard helps you copy certificates, certificate trust lists, and certificate revocation lists from your disk to a certificate store.

A certificate, which is issued by a certification authority, is a confirmation of your identity and contains information used to protect data or to establish secure network connections. A certificate store is the system area where certificates are kept.

To continue, click Next.

14. When you complete the steps in the Import Wizard, “The Import was successful” message appears on your screen.

Installing a certificate on LCS/OCS servers

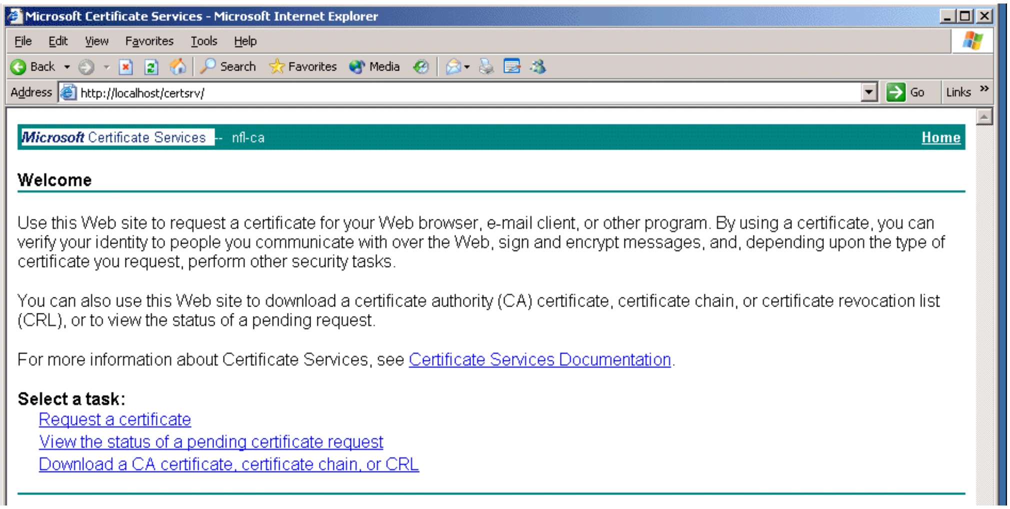
The section covers the steps for installing the certificate file on an LCS/OCS server.

1. From the LCS/OCS Server, go to the HTTP location where you are running the Microsoft Certificate Authority. For example:

   http://localhost/certserv

   The Microsoft Certificate Services window appears.

2. If you have not already done so, log on as **Administrator** of the CA.


Microsoft Certificate Services - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost/certsrv/
Go Links

Microsoft Certificate Services -- nfl-ca Home

Welcome

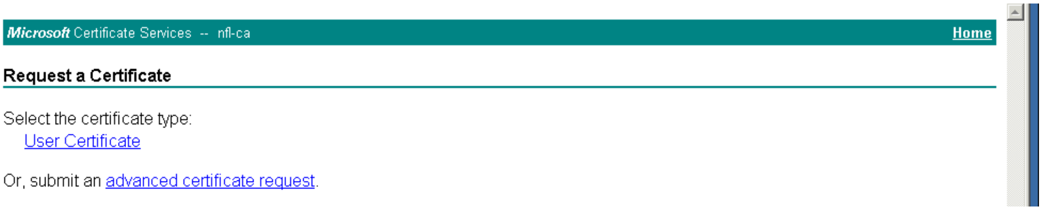
Use this Web site to request a certificate for your Web browser, e-mail client, or other program. By using a certificate, you can verify your identity to people you communicate with over the Web, sign and encrypt messages, and, depending upon the type of certificate you request, perform other security tasks.

You can also use this Web site to download a certificate authority (CA) certificate, certificate chain, or certificate revocation list (CRL), or to view the status of a pending request.

For more information about Certificate Services, see Certificate Services Documentation.

Select a task:
Request a certificate
View the status of a pending certificate request
Download a CA certificate, certificate chain, or CRL

3. Select **Request a certificate** to display the Request a Certificate window.



Microsoft Certificate Services -- nfl-ca Home

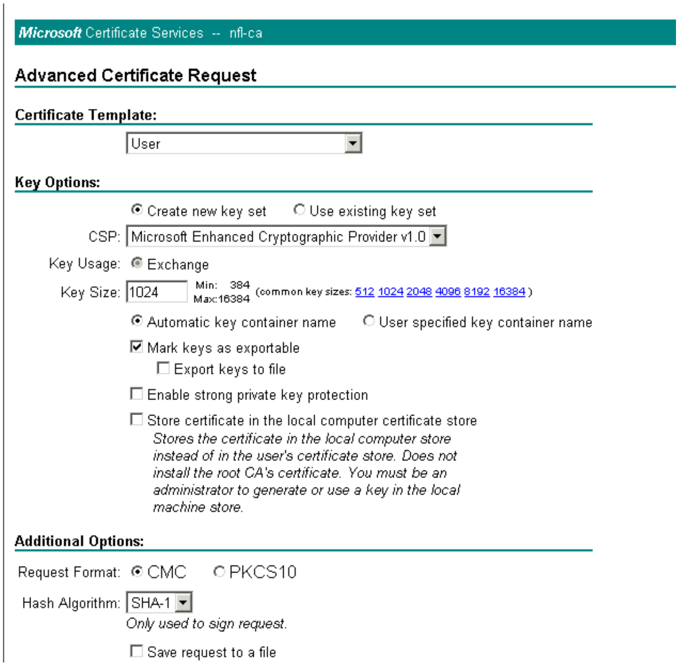
Request a Certificate

Select the certificate type:
User Certificate

Or, submit an advanced certificate request.

4. Select **advanced certificate request->Create and submit a request to this CA**.

The Advanced Certificate Request window appears.



Microsoft Certificate Services -- nfl-ca

Advanced Certificate Request

Certificate Template:

User

Key Options:

● Create new key set ○ Use existing key set

CSP: Microsoft Enhanced Cryptographic Provider v1.0

Key Usage: ● Exchange

Key Size: 1024 Min: 384 Max:16384 (common key sizes: 512 1024 2048 4096 8192 16384)

● Automatic key container name ○ User specified key container name

☑ Mark keys as exportable

□ Export keys to file

□ Enable strong private key protection

□ Store certificate in the local computer certificate store

Stores the certificate in the local computer store instead of in the user's certificate store. Does not install the root CA's certificate. You must be an administrator to generate or use a key in the local machine store.

Additional Options:

Request Format: ● CMC ○ PKCS10

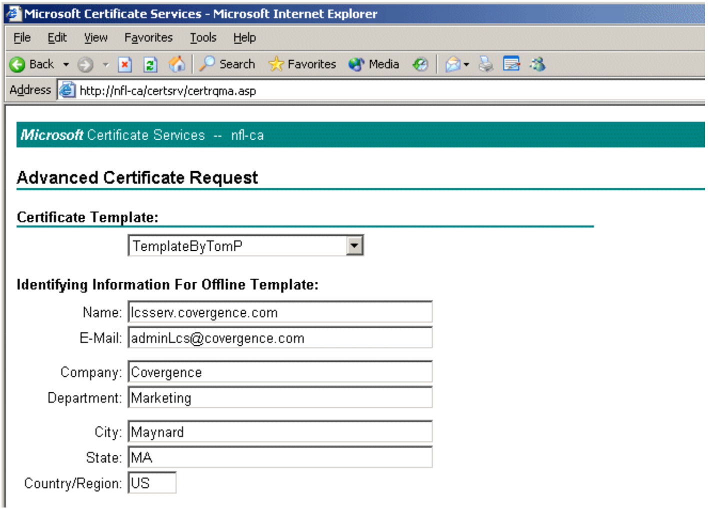
Hash Algorithm: SHA-1

Only used to sign request.

□ Save request to a file

5. Under the “Certificate Template” section, there is a pull-down menu with selections for identifying the LCS/OCS server where the certificate will be installed. If an offline certificate was previously created, the name of the template should appear as one of the selections. Select the template from the pull-down menu.

6. Under the “Identifying Information of Offline Template” section, the “Name” field should contain the fully-qualified domain name for the LCS/OCS server, such as *lcsserv.acmepacket.com*, as shown in.


Microsoft Certificate Services - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address

http://nfl-ca/certsrv/certrqma.asp

Microsoft Certificate Services -- nfl-ca

Advanced Certificate Request

Certificate Template:

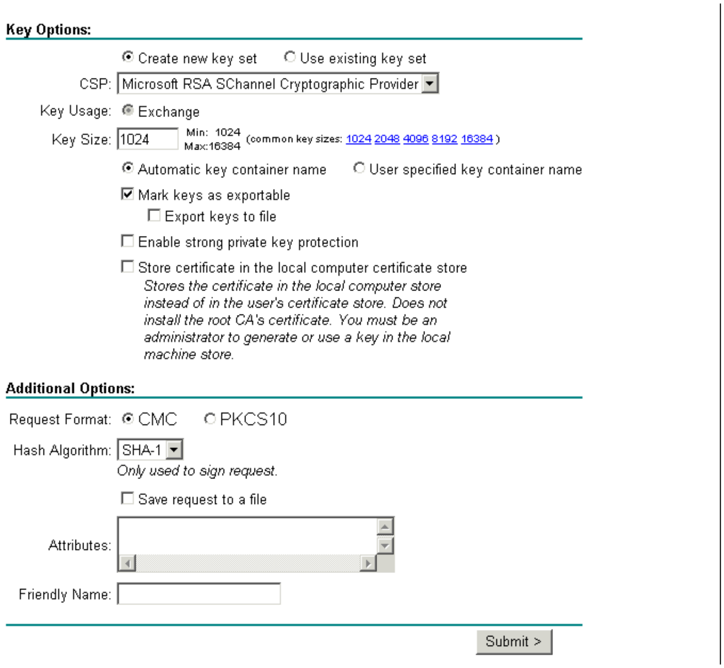
TemplateByTomP

Identifying Information For Offline Template:

Name:	lcsserv.covergence.com
E-Mail:	adminLcs@covergence.com
Company:	Covergence
Department:	Marketing
City:	Maynard
State:	MA
Country/Region:	US

7. Under “Key Options”, select **Store certificate in the local computer certificate store**.

8. Under “Additional Options”, optionally enter a text string in the **Friendly Name** field.



Key Options:

● Create new key set ○ Use existing key set

CSP: Microsoft RSA SChannel Cryptographic Provider

Key Usage: ● Exchange

Key Size: 1024 Min: 1024 Max:16384 (common key sizes: 1024 2048 4096 8192 16384)

● Automatic key container name ○ User specified key container name

☑ Mark keys as exportable

□ Export keys to file

□ Enable strong private key protection

□ Store certificate in the local computer certificate store

Stores the certificate in the local computer store instead of in the user's certificate store. Does not install the root CA's certificate. You must be an administrator to generate or use a key in the local machine store.

Additional Options:

Request Format: ● CMC ○ PKCS10

Hash Algorithm: SHA-1

Only used to sign request.

□ Save request to a file

Attributes:

Friendly Name:

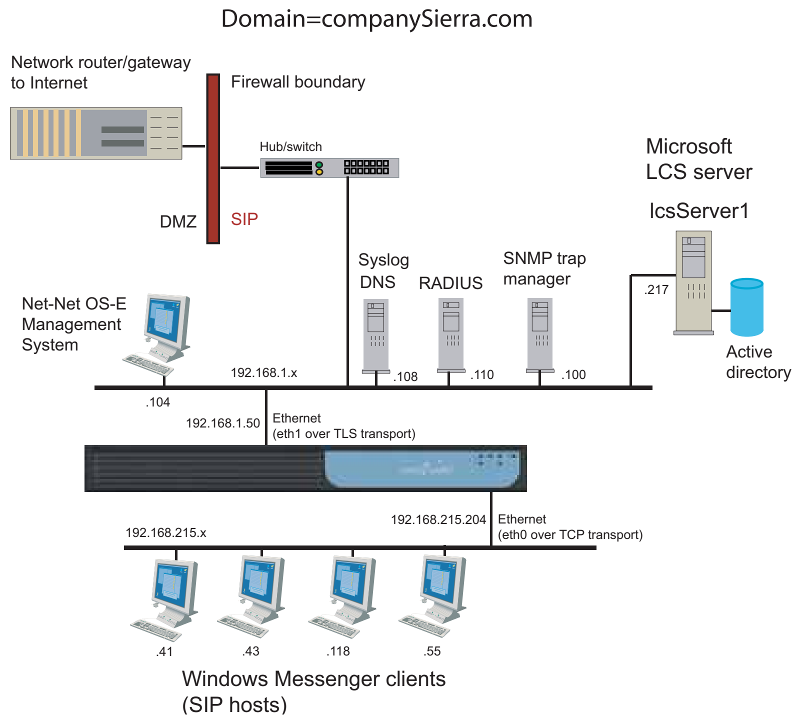
9. Click **Submit** and answer **Yes** to any security questions.
10. Select **Install this certificate**. You will receive a message when the certificate has been successfully installed.

Exporting certificates to AA-SBC

This section covers the steps for acquiring a certificate from a SIP server, such as the Microsoft LCS. This will allow you to run SIP sessions over a secure TLS transport between AA-SBC and the LCS server. To get a certificate from the LCS server, you need to log on to the certificate authority (CA) and request a certificate.

The above image illustrates a sample network where a certificate is imported from the LCS server named *lcsServer1* to the AA-SBC platform. The server is configured as follows:

```
NNOS-E> config vsp
config vsp> config enterprise
config enterprise> config servers
config servers> config lcs lcsServer1
config lcs lcsServer1> set admin enabled
config lcs lcsServer1> set domain companySierra.com
config lcs lcsServer1> set server-type lcs-2005
config lcs lcsServer1> set directory
vsp\enterprise\directories\active-directory companySierra
config lcs lcsServer1> set peer-identity
    sip:lcsServer1.companySierra.com
config lcs lcsServer1> set directory
    vsp\enterprise\directories\active-directory
    lcsServer1.companySierra
config lcs server1> set server-type lcs-2005

config lcs lcsServer1> config server-pool
config server-pool> config server lcsServer1
Creating 'server lcsServer1'
config server lcsServer1> set host 192.168.1.217
config server lcsServer1> set transport TCP
```



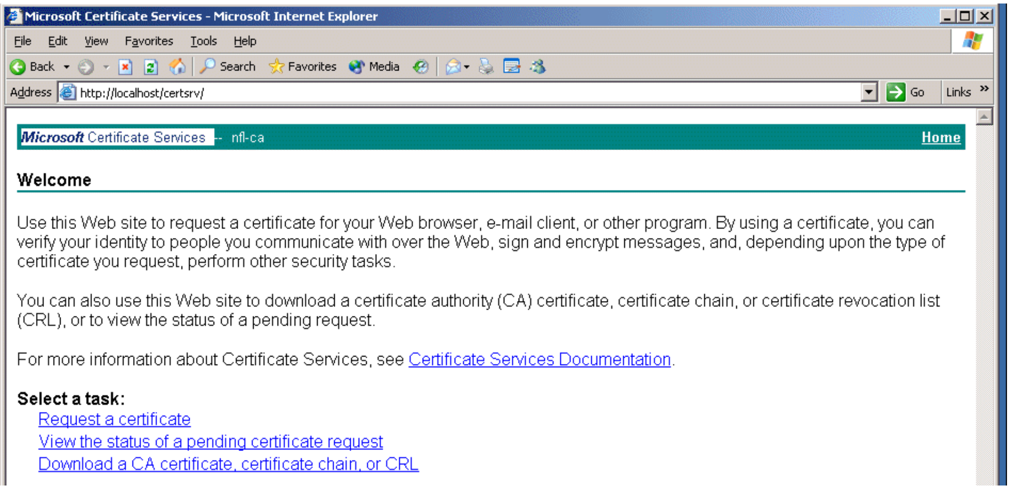
The steps for completing the certificate process from a Microsoft LCS server are as follows:

1. From the LCS Server, go to the HTTP location where you are running the Microsoft Certificate Authority. For example:

   http://localhost/certserv

   The Microsoft Certificate Services window appears.

2. If you have not already done so, log on as **Administrator** of the CA.



Microsoft Certificate Services - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost/certsrv/
Go Links

Microsoft Certificate Services -- nfl-ca Home

Welcome

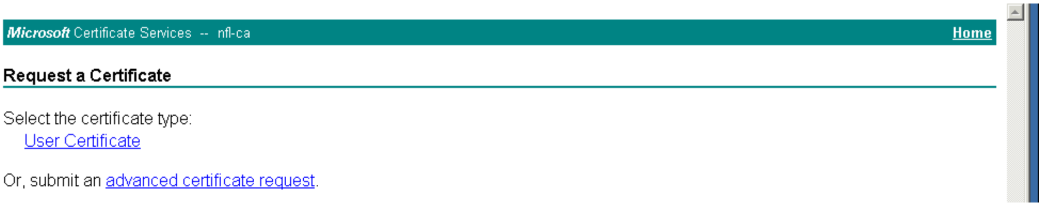
Use this Web site to request a certificate for your Web browser, e-mail client, or other program. By using a certificate, you can verify your identity to people you communicate with over the Web, sign and encrypt messages, and, depending upon the type of certificate you request, perform other security tasks.

You can also use this Web site to download a certificate authority (CA) certificate, certificate chain, or certificate revocation list (CRL), or to view the status of a pending request.

For more information about Certificate Services, see Certificate Services Documentation.

Select a task:
Request a certificate
View the status of a pending certificate request
Download a CA certificate, certificate chain, or CRL

3. Select **Request a certificate** to display the Request a Certificate window.



Microsoft Certificate Services -- nfl-ca Home

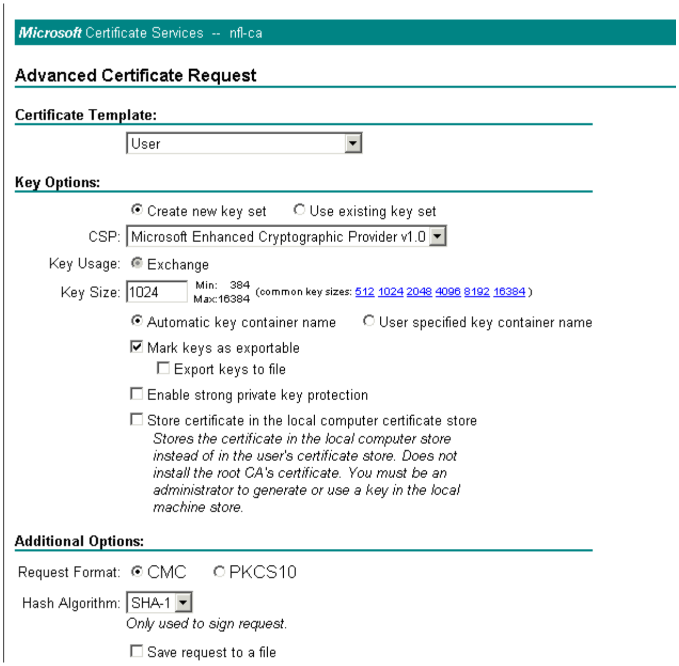
Request a Certificate

Select the certificate type:
User Certificate

Or, submit an advanced certificate request.

4. Select **advanced certificate request->Create and submit a request to this CA**.

   The Advanced Certificate Request window appears.



Microsoft Certificate Services -- nfl-ca

Advanced Certificate Request

Certificate Template:

User

Key Options:

● Create new key set ○ Use existing key set

CSP: Microsoft Enhanced Cryptographic Provider v1.0

Key Usage: ● Exchange

Key Size: 1024 Min: 384 Max:16384 (common key sizes: 512 1024 2048 4096 8192 16384)

● Automatic key container name ○ User specified key container name

☑ Mark keys as exportable

□ Export keys to file

□ Enable strong private key protection

□ Store certificate in the local computer certificate store

Stores the certificate in the local computer store instead of in the user's certificate store. Does not install the root CA's certificate. You must be an administrator to generate or use a key in the local machine store.

Additional Options:

Request Format: ● CMC ○ PKCS10

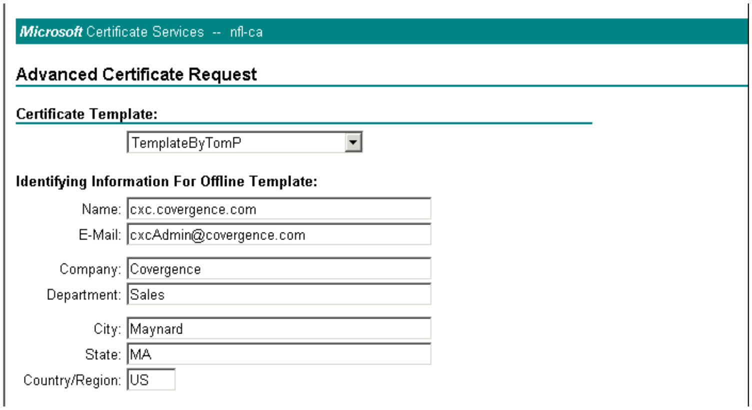
Hash Algorithm: SHA-1

Only used to sign request.

□ Save request to a file

5. Under the “Certificate Template” section, there is a pull-down menu with selections for identifying the LCS server from which the certificate will be exported to the AA-SBC platform. If an offline certificate was previously created, the name of the template should appear as one of the selections. Select the template from the pull-down menu.

6. Under the “Identifying Information of Offline Template” section, the “Name” field should contain the fully-qualified domain name for the AA-SBC platform, such as *acmepacket.com*, as shown in.



Microsoft Certificate Services -- nfl-ca

Advanced Certificate Request

Certificate Template:

TemplateByTomP ▾

Identifying Information For Offline Template:

Name: cxc.covergence.com

E-Mail: cxcAdmin@covergence.com

Company: Covergence

Department: Sales

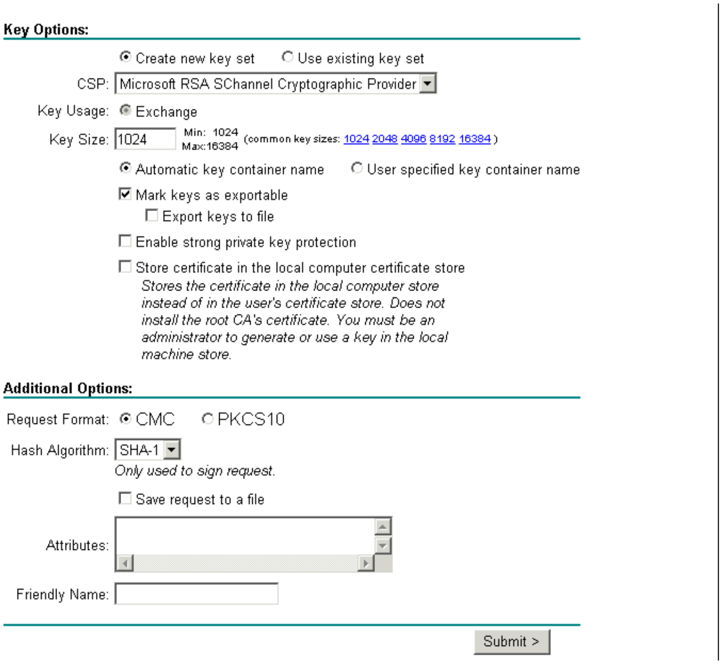
City: Maynard

State: MA

Country/Region: US

7. Under “Key Options”, check the boxes **Store certificate in the local computer certificate store** and **Mark key as exportable.**

8. Under “Additional Options”, optionally enter a text string in the **Friendly Name** field.



Key Options:

● Create new key set ○ Use existing key set

CSP: Microsoft RSA SChannel Cryptographic Provider ▾

Key Usage: ● Exchange

Key Size: 1024 Min: 1024 Max:16384 (common key sizes: 1024 2048 4096 8192 16384)

● Automatic key container name ○ User specified key container name

☑ Mark keys as exportable

☐ Export keys to file

☐ Enable strong private key protection

☐ Store certificate in the local computer certificate store

Stores the certificate in the local computer store instead of in the user's certificate store. Does not install the root CA's certificate. You must be an administrator to generate or use a key in the local machine store.

Additional Options:

Request Format: ● CMC ○ PKCS10

Hash Algorithm: SHA-1 ▾

Only used to sign request.

☐ Save request to a file

Attributes:

Friendly Name:

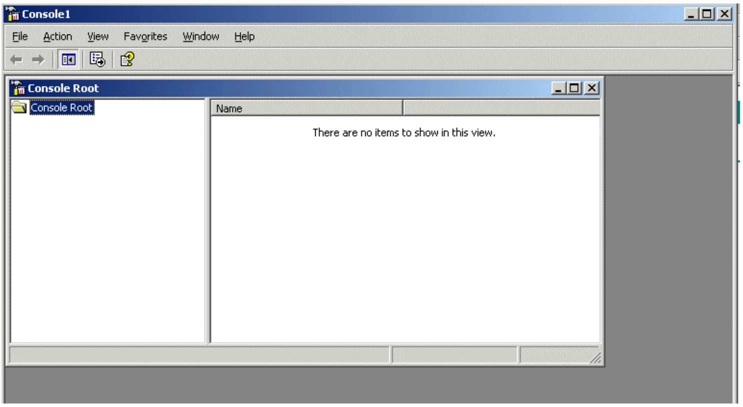
9. Click **Submit** and answer **Yes** to any security questions.

10. Select **Install this certificate**.

11. Click the **Submit** button.

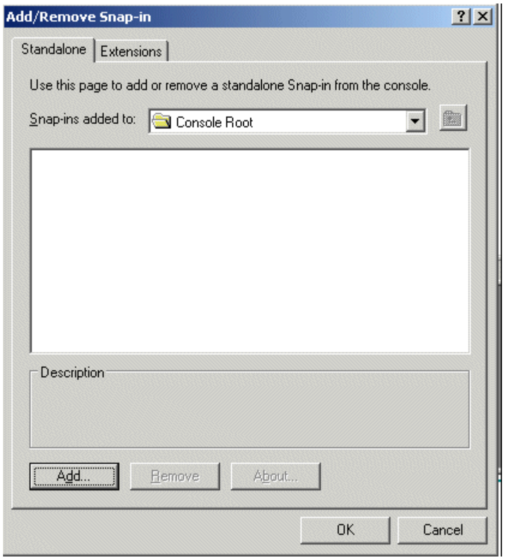
12. Select **Install this certificate**. You will receive a message when the certificate has been successfully installed.

13. At the **Start** button, click **Run**, then enter **mmc** on the command line. Click **OK** to display the Console Root window.



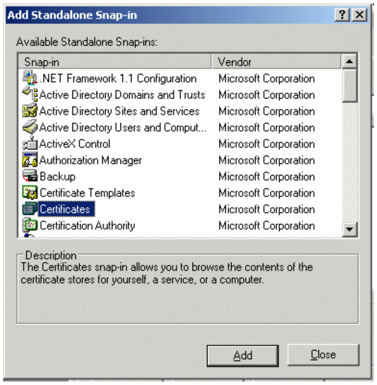
There are no items to show in this view.

14. From the Console Root window, go to the pull-down menu and select **File ->Add/ Remove Snap-in.** The Add/Remove Snap-in window appears.



Use this page to add or remove a standalone Snap-in from the console.

15. From the Add/Remove Snap-in window, click **Add**, then select **Certificates** from the list of Snap-in options in the Add Standalone Snap-in window.



Add Standalone Snap-in

Available Standalone Snap-ins:

Snap-in	Vendor
.NET Framework 1.1 Configuration	Microsoft Corporation
Active Directory Domains and Trusts	Microsoft Corporation
Active Directory Sites and Services	Microsoft Corporation
Active Directory Users and Comput...	Microsoft Corporation
ActiveX Control	Microsoft Corporation
Authorization Manager	Microsoft Corporation
Backup	Microsoft Corporation
Certificate Templates	Microsoft Corporation
Certificates	Microsoft Corporation
Certification Authority	Microsoft Corporation

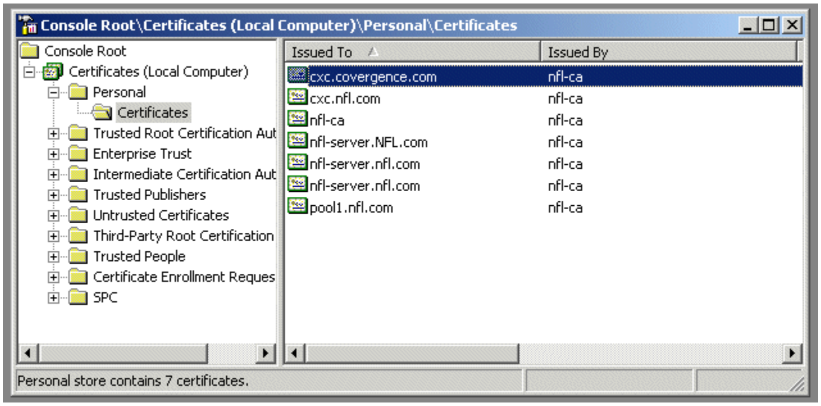
Description

The Certificates snap-in allows you to browse the contents of the certificate stores for yourself, a service, or a computer.

16. Click **Add->Computer Account->Next->Local computer.** Click **Finish**.

17. Click **Close,** then **OK**, to return to the Console Root window.

18. Double-click **Certificates**, expand the **Personal** folder. Click the **Certificates** folder. You should see the certificate you just created.



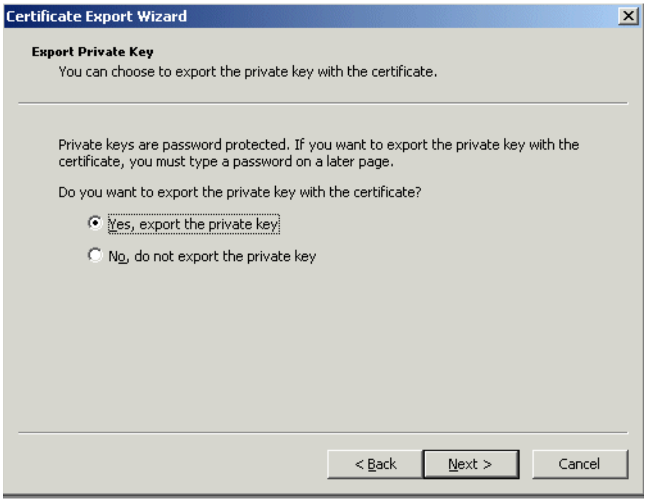
Console Root\Certificates (Local Computer)\Personal\Certificates

- Console Root
  - Certificates (Local Computer)
    - Personal
      - Certificates
    - Trusted Root Certification Aut
    - Enterprise Trust
    - Intermediate Certification Aut
    - Trusted Publishers
    - Untrusted Certificates
    - Third-Party Root Certification
    - Trusted People
    - Certificate Enrollment Reques
    - SPC

Issued To	Issued By
cxc.covergence.com	nfl-ca
cxc.nfl.com	nfl-ca
nfl-ca	nfl-ca
nfl-server.NFL.com	nfl-ca
nfl-server.nfl.com	nfl-ca
nfl-server.nfl.com	nfl-ca
pool1.nfl.com	nfl-ca

Personal store contains 7 certificates.

19. Right-click the certificate Select **All Tasks->Export** to start the Certificate Export Wizard. Click **Next** to display the Export Private Key window.



Certificate Export Wizard

Export Private Key

You can choose to export the private key with the certificate.

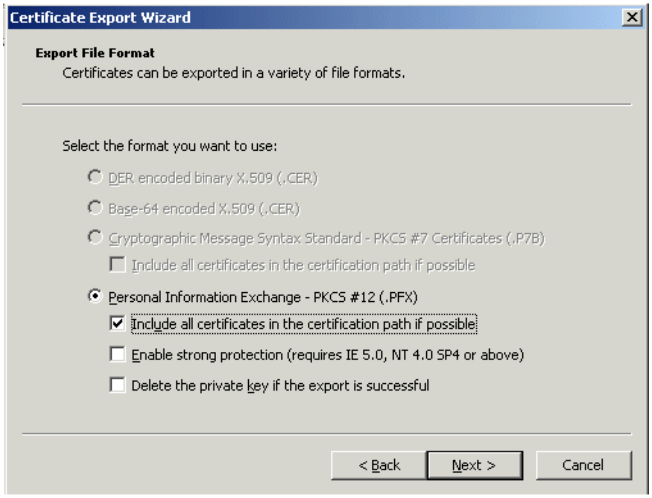
Private keys are password protected. If you want to export the private key with the certificate, you must type a password on a later page.

Do you want to export the private key with the certificate?

● Yes, export the private key

○ No, do not export the private key

20. Select **Yes, export private key**. Click **Next** to display the Export File Format window.



Certificate Export Wizard

Export File Format

Certificates can be exported in a variety of file formats.

Select the format you want to use:

○ DER encoded binary X.509 (.CER)

○ Base-64 encoded X.509 (.CER)

○ Cryptographic Message Syntax Standard - PKCS #7 Certificates (.P7B)

□ Include all certificates in the certification path if possible

● Personal Information Exchange - PKCS #12 (.PFX)

☑ Include all certificates in the certification path if possible

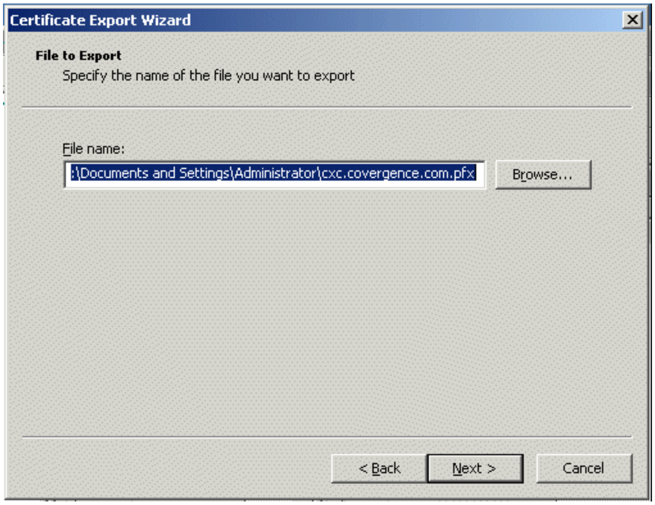
□ Enable strong protection (requires IE 5.0, NT 4.0 SP4 or above)

□ Delete the private key if the export is successful

21. Uncheck **Enable strong protection** and check **Include all certificates in the certificate path if possible**.

22. Click **Next**. Enter and confirm a password of your choice. Click **Next**.

23. In the File to Export window, click **Browse** to locate the file. In the example, the certificate is named *acmepacket.com.pfx*.



Certificate Export Wizard

File to Export

Specify the name of the file you want to export

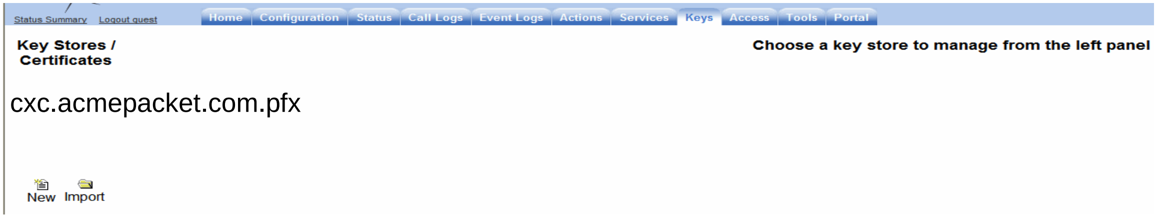
File name:

:\Documents and Settings\Administrator\cxc.covergence.com.pfx

24. Click **Next** to display the Export settings that you chose, then click **Finish**. You should receive the message “The export was successful.”

25. Move the certificate to the AA-SBC platform and store it under the cxc/certs file. You can use the AA-SBC Management System and browse for the certificate, or use PSCP or SFTP to place transfer it to the AA-SBC platform, as described in the section “Importing certificates using file transfer.”

26. From the AA-SBC Management System **Key**s tab, verify that the certificate is in the key store as follows:



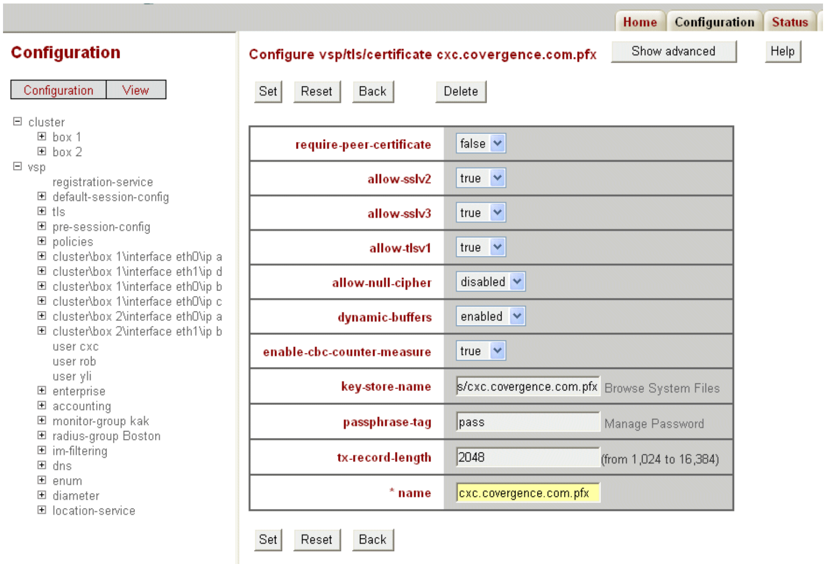
Key Stores / Certificates

Choose a key store to manage from the left panel

cxc.acmepacket.com.pfx

27. Create a configuration entry for the certificate in VSP TLS configuration.

    Type in the certificate path. In this example, the path is *cxc/certs/acmepacket.com.pfx*.



Configure vsp/tls/certificate cxc.covergence.com.pfx

require-peer-certificate	false
allow-sslv2	true
allow-sslv3	true
allow-tlsv1	true
allow-null-cipher	disabled
dynamic-buffers	enabled
enable-cbc-counter-measure	true
key-store-name	s/cxc.covergence.com.pfx
passphrase-tag	pass
tx-record-length	2048 (from 1,024 to 16,384)
* name	cxc.covergence.com.pfx

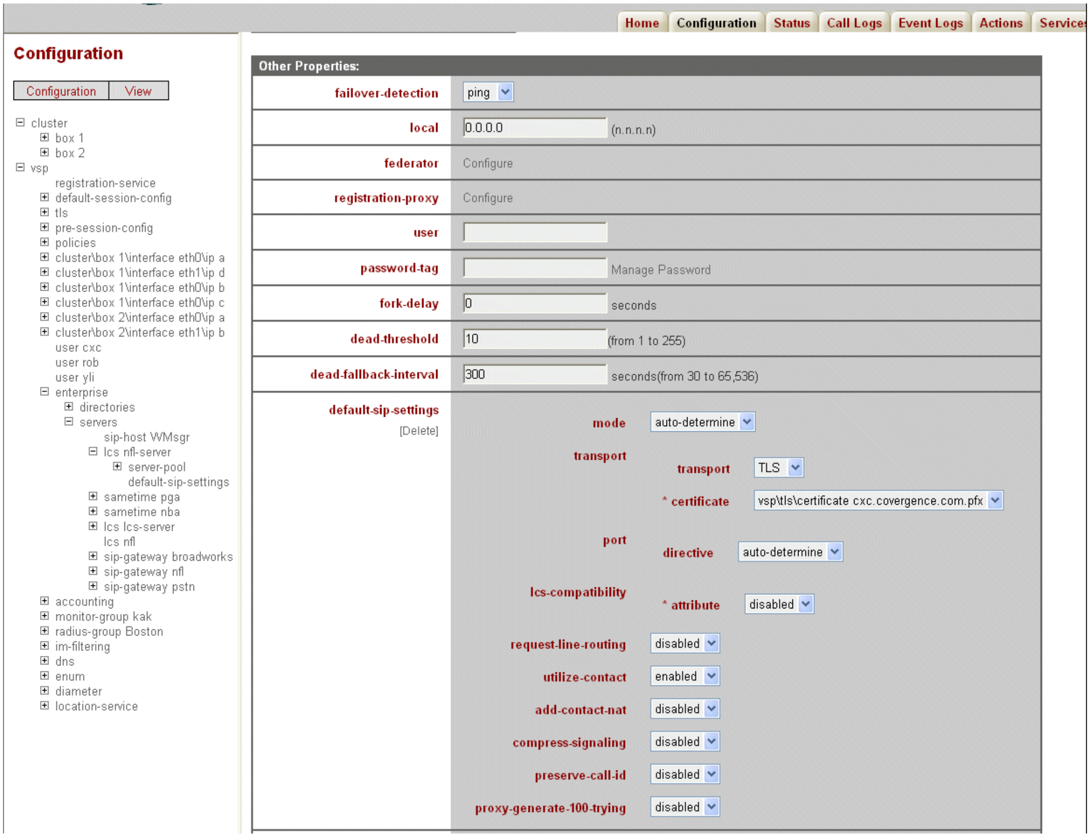
28. Using the CLI, enter the **show certificates -v** command at the prompt. Make sure the certificate is working with the State field reporting OK.

```
NNOS-E> show certificates -v

------------------------------------------------------------------
Cert certificate status for '<Outgoing>'...
------------------------------------------------------------------
Name:                   <Default Outgoing>
Record ID:              -1
State:                  OK
File name:
File stamp:             00000000
File stamp string:
File format:            None
Certificate name:
  Issuer:
  Subject:
  Common name:
  DNS name:
  Not before:
  Not after:
Private key OK:         No
Allow SSLv2:            No
Allow SSLv3:            Yes
Allow TLSv1:            Yes
Allow null cipher:  Disabled
Dynamic buffers     Enabled
CBC counter-meas:   Yes
Cipher spec:        ALL:!ADH:!LOW:!EXP:!MD5:@STRENGTH
Tx record length:   2048 bytes
Peer cert verify:   None
  Required name:
  Default CA:       Yes
  Custom CA:        <None>
  Default CRL:      Yes
  Custom CRL:       <None>
```

29. Configure the **default-sip-settings** properties. If you are using the CLI, follow the session below.

```
NNOS-E> config vsp enterprise servers lcs lcs-server
config lcs lcsServer> config default-sip-settings
config default-sip-settings> set mode auto-determine
config default-sip-settings> set lcs-compatibility disabled
config default-sip-settings> set transport tls "vsp\tls\certificate
cxc.acmepacket.com.pfx
config default-sip-setting> return
config lcs lcsServer> return
config servers>
```

The following image shows the default-sip-setting page in the AA-SBC Management System.



Other Properties:

failover-detection	ping
local	0.0.0.0 (n.n.n.n)
federator	Configure
registration-proxy	Configure
user	
password-tag	Manage Password
fork-delay	0 seconds
dead-threshold	10 (from 1 to 255)
dead-fallback-interval	300 seconds(from 30 to 65,536)

default-sip-settings [Delete]

mode		auto-determine
transport	transport	TLS
	* certificate	vsp\tls\certificate cxc.covergence.com.pfx
port	directive	auto-determine
lcs-compatibility	* attribute	disabled
request-line-routing		disabled
utilize-contact		enabled
add-contact-nat		disabled
compress-signaling		disabled
preserve-call-id		disabled
proxy-generate-100-trying		disabled

Importing certificates using file transfer

If you are not using the AA-SBC Management System, as covered in the previous section, copy the certificate that you receive from the CA to the AA-SBC platform using a compatible file transfer mechanism, such as PuTTY Secure Copy (PSCP). If you have the file on a local network PC, use PSCP to move the file to a directory path on the AA-SBC platform.

CLI session

The following example PSCP command copies the certificate file named **`cxc.acmepacket.com.pfx`** from the PC root directory to the AA-SBC platform at IP address **`192.168.215.100`** in the directory **`/cxc/certs/cert.acmepacket.com.pfx`**.

```
C:\ pscp -l root -pw sips -P 2200 cxc.acmepacket.com.pfx
   192.168.215.100:/cxc/certs/cxc.acmepacket.com.pfx
```

Configuring the certificate on AA-SBC

If you are using the CLI, and once you have imported the certificate to a directory on the AA-SBC platform, configure the certificate settings that control how AA-SBC uses the certificate.

CLI session

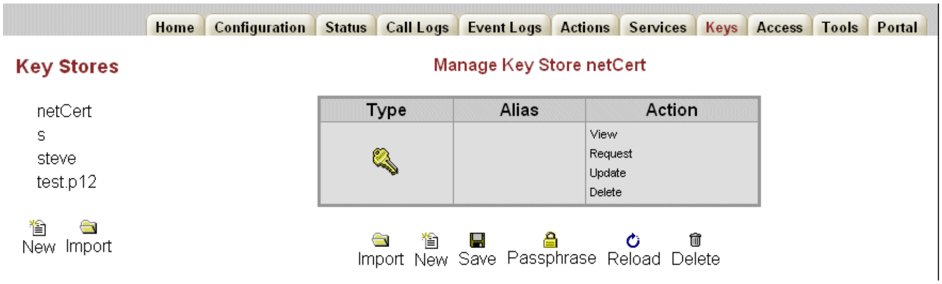
The following CLI session sets the directory and certificate file name path, specifies the passphrase, and whether to allow SSL Version 2 operability.

```
NNOS-E> config vsp
config vsp> config tls
config tls> config certificate cxc.acmepacket.com.pfx
config certificate cxc.acmepacket.com.pfx> set allow-sslv2 true
config certificate cxc.acmepacket.com.pfs> set certificate-file /cxc/
   certs/cxc.acmepacket.com.pfx
config certificate cxc.acmepacket.com.pfx> set passphrase-tag pass
```

By default, AA-SBC only supports SSLv3 or TLSv1. If you require SSLv2 for interoperability, set this property **true**. Specify the passphrase-tag associated with the certificate file. Use this property if the certificate file is encrypted to have its private key information protected. This passphrase tag must match the string with which the certificate was encrypted.

Managing cryptographic keys

The AA-SBC Managment System allows you to manage cryptographic keys before you configure them on the AA-SBC platform. The **Keys** tab allows you to add and maintain the necessary key data. The following image displays the key store management page.



Key Stores

netCert
s
steve
test.p12

Manage Key Store netCert

Type	Alias	Action
		View Request Update Delete

A key store is the file that maintains the certificate and/or keys that can be used by AA-SBC. From the **Keys** tab, you can manage and install the key stores. The following table describes each button or link available in the key store management window.

Button	Action
New	Creates a new key store, a named container for key and certificate data, on the local device. Enter a key store name and assign a pass phrase
Import (key store)	Imports a key store, a named container and its associated data, onto the local device. You can either supply a file path and, optionally, a destination name for the key store or you can browse for a key store saved on your network.
The following are available once you are in a key store	
Import (keys/certificates)	Uploads certificates and/or keys into the highlighted key store. Enter the file path or browse for the data you wish to import and, if desired, an alias. If a password is required to access the data, enter it here.
Save	Saves the currently open key store. Until a key store is saved, the it is not available for use.
Passphrase	Allows you to change the pass phase for a key store. Because you had to enter the current pass phrase to have editing access to the key store, the dialog box only prompts for the new pass phrase and a confirmation.
Reload	Reloads the current key store and applies any changes that may have occurred since the key store was first opened.
Delete	Deletes the selected key store off the box.

7. Configuring SIP servers, directories, and federations

About this chapter

This chapter provides information on configuring SIP servers, directories of SIP-addressable users, SIP hosts, and federations.

Servers, directories, and federations

Enterprise SIP servers, directories, and federations are components that interoperate with the AA-SBC platform.

While SIP servers provide the SIP applications and endpoints for SIP sessions, directories store information about the users who are available to participate in SIP sessions. Whenever a SIP client attempts to contact a call recipient using a SIP INVITE message, either within the enterprise domain or across the Internet, the AA-SBC platform performs a “lookup” in a configured directory. If a directory entry exists for the call recipient, and AA-SBC locates the recipient using entries in the location service database, AA-SBC connects the call participants. Each SIP call participant then communicates over the SIP session using the SIP application provided by the enterprise server(s), such as Microsoft LCS, SIP phones connected to SIP PBXs, etc.

The AA-SBC platform applies a “default” policy to all SIP sessions, or if configured, a set of policies (rules and conditions) based on the type of call session, users participating in the call, and other policy settings. Policies are the mechanisms that provide the detailed security filtering on SIP sessions.

Federations are groupings of SIP servers that allow users of those servers to communicate. Federations provide the ability to connect across organizations, such as one LCS deployment to another or LCS-to-IBM Sametime. 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. The AA-SBC platform acts as the device in the middle (an access proxy for LCS, a SIP connector for Sametime, for example).

Interoperating with SIP servers

The AA-SBC platform is designed to interoperate with SIP servers, hosted SIP applications, and SIP PBX equipment for voice over IP applications. AA-SBC operates as a SIP proxy or back-to-back user agent (B2BUA) between the SIP clients and the SIP servers.

Before you configure the policies that manage SIP sessions, you will need to install the SIP servers in your enterprise network. Refer to the following appendixes in the *Net-Net OS-E – System Administration Guide* for the information that applies to the servers used in your network configuration.

- Appendix A, "Interoperating with Microsoft OCS to IBM Lotus Sametime 7.5,"
- Appendix B, "Interoperating with Microsoft OCS to IBM Lotus Sametime 6.5/ 7.0,"

Configuring the SIP server pools

SIP servers are the endpoints that provide SIP-enabled real-time communication collaboration services for SIP sessions. Enterprise SIP communications servers include:

- IBM Lotus Sametime
- Microsoft Live Communications Server 2005
- Nortel MCS
- Avaya IP telephony PBX
- SIP hosts (generic SIP source, such as Windows Messenger clients, or destination servers)
- SIP connections
- SIP gateways (application servers, registrars, PSTN gateways, open-source Asterisk PBX)
- Domain Naming System (DNS) server groups

SIP servers allows enterprises to support features such as IP PBX-hosted VoIP services, Instant messaging (IM) between SIP users, mobile services, and presence-based applications.

A server pool is a group of enterprise servers, with all servers in the group having the same characteristics, and from the same vendor so that a matching dial-plan can apply to all servers in the pool. Servers configured in the pool can perform traffic balancing with the other servers.

IBM Lotus Sametime

The IBM Lotus Instant Messaging and Web Conferencing (Sametime) is IBM's real-time collaboration platform. It is used to manage the flow of instant messages, streaming audio and video, shared applications, and white board sessions. Sametime provides instant access to people and information through integrated presence awareness and brings together centralized and geographically dispersed participants.

CLI session

The following CLI session shows the `sametime` object hierarchy.

```
NNOS-E> config
config> config vsp enterprise servers
config servers> config sametime name
Creating ‘sametime name’
config sametime name>
```

Microsoft LCS

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.

CLI session

The following CLI session shows the **lcs** object hierarchy and a basic configuration.

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

```
config servers> config lcs lcsServer1
Creating ‘lcs lcsServer1’
config lcs lcsServer1> set admin enabled
config lcs lcsServer1> set domain lcsServer1.company.com
config lcs lcsServer1> set peer-identity sip:lcsServer1.company.com
config lcs lcsServer1> set directory
   vsp\enterprise\directories\active-directory lcsServer1.company
config lcs lcsserver1> set server-type lcs-2005

config lcs lcsServer1> config server-pool
config server-pool> config server lcsServer1
Creating 'server lcsServer1'
config server lcsServer1> set host 192.168.1.217
config server lcsServer1> set transport TCP
```

Nortel MCS

The Nortel Multimedia Communication Server (MCS) delivers SIP-based multimedia and collaborative applications to enterprises. Using industry-standard protocols, it enables businesses to augment existing voice and data infrastructures with advanced IP-based capabilities, such as multimedia (video conferencing and calling, picture caller ID); collaboration (conferencing, white boarding, file exchange, co-Web browsing); personalization (call screening, call logs, call management and routing - find me, follow me); presence and instant messaging.

CLI session

The following CLI session shows the **mcs** object hierarchy and a sample configuration.

```
NNOS-E> config vsp enterprise servers
config servers> config mcs mcsServer1
Creating ‘mcs lcsServer1’
config mcs mcsServer1> set admin enabled
config mcs mcsServer1> set domain mcsServer1.company.com
config mcs mcsServer1> set peer-identity sip:mcsServer1.company.com
config mcs mcsServer1> set directory
   vsp\enterprise\directories\active-directory mcsServer1.company
config mcs mcsServer1> set server-type nortel-mcs
config mcs mcsServer1> config server-pool
config server-pool> config server mcsServer1
Creating 'server mcs Server1'
config server lcsServer1> set host 192.168.1.218
config server lcsServer1> set transport TCP
```

Avaya PBX

Avaya's 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, CSS integrates SIP registrar, proxy, presence, instant message gateway, and instant messaging to provide full communications in a multivendor environment.

CLI session

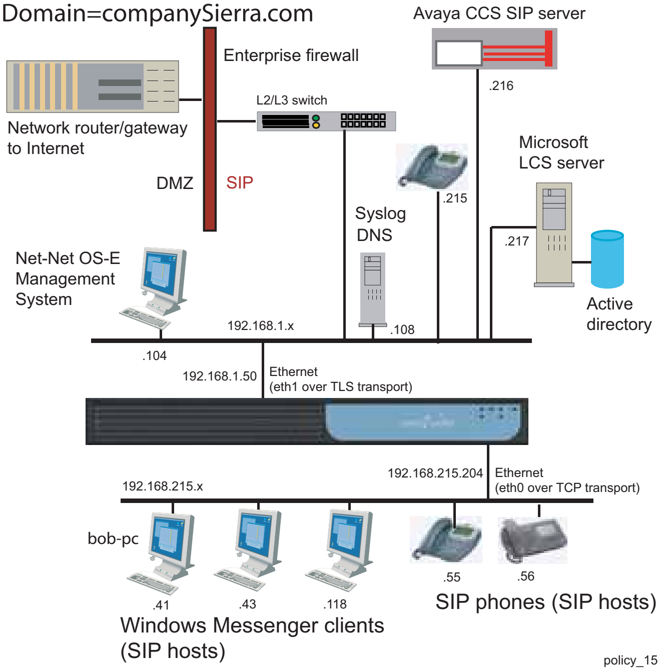
The following CLI session shows the **avaya** object hierarchy.

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

SIP hosts

A SIP host is usually a SIP client, such as a PC running Windows Messenger, that establishes SIP sessions with a SIP server. SIP hosts can also include SIP phones, as well as PCs and workstations that are running hosted SIP applications with a single network domain.

The following image illustrates a sample network using Window Messenger and SIP phones as hosts.



CLI session

The following CLI session configures the SIP host at IP address 192.168.215.41, sets the domain name to which the sip host belongs, and sets the type, IP address and host domain name, and specifies that active directory providing naming services to this host.

```
config> config vsp enterprise servers
config servers> config sip-host 192.168.215.41
config sip-host 192.168.215.41> set admin enabled
config sip-host 192.168.215.41> set domain companySierra.com
config sip-host 192.168.215.41> set server-type windows-messenger
...192.168.215.41> set loop-detection tight
...192.168.215.41> set directory
   vsp\enterprise\directories\active-directory\companySierra
```

SIP gateways

SIP gateways include SIP application servers, SIP registrars, PSTN gateways, and open-source Asterisk PBXs.

You can configure AA-SBC to allow enterprises to continue call operations even if a service provider server is busy or down. You do this by configuring a public-switched telephone network (PSTN) gateway.

AA-SBC normally forwards calls to a service provider application server. If the server has failed, and if AA-SBC has location information for the provider, it forwards calls locally. Otherwise, AA-SBC forwards calls to a PSTN gateway, configured using the **sip-gateway** server object.

AA-SBC detects provider failure for one or more of the following reasons:

- No route to the provider.
- Registration failures.
- Outbound call from AA-SBC to the provider reaches the maximum number of retransmissions.

Use the **sip-gateway failover-detection** property to control how AA-SBC should behave if a SIP server failure is detected.

CLI session

The following CLI session shows the **sip-gateway** object hierarchy. Specify the SIP URI for the gateway, in the format *sip:gatewayIdentity*. For example, *sip:server@provider1.com*.

```
config> config vsp enterprise
config enterprise> config servers
config servers> config sip-gateway pstn
config sip-gateway pstn> config server-pool
config server-pool> config server sip:pstn1@provider1.com
config server sip:pstn1@provider1.com> return
config server-pool> config server sip:pstn2@provider1.com
config server sip:pstn2@provider1.com> return
config sip-gateway pstn> set user cxc
config pstn-gateway pstn> set password abc123xyz
config pstn-gateway pstn> return
config servers> return
config enterprise> return
```

SIP registrars

SIP registrars are servers that handle SIP REGISTER requests. The AA-SBC platform functions as a local SIP registrar, where AA-SBC accepts SIP REGISTER requests on behalf of SIP clients. AA-SBC also forwards and delegates SIP REGISTER requests to one or more registration peers, allowing the SIP clients and the registrars to communicate directly, where SIP user address-of-record updates are exchanged.

Use the **sip-gateway** object to configure the peer registrars to which AA-SBC forwards REGISTER packets. The domain**,** peer-identity, and domain-alias properties define the addresses-of-record for the SIP registrar, as well as how strictly to enforce the matching of domains, aliases, and IP addresses contained in the To: header in SIP REGISTER messages.

The following CLI configures a registration peer and the associated contact information, and sets the communication properties for REGISTER sessions between AA-SBC and the registrar peer.

CLI session

```
config> config vsp enterprise
config vsp enterprise> config servers
config servers> config sip-gateway broadworks
config sip-gateway broadworks> set domain as.broadworks.net
config sip-gateway broadworks> set domain-alias broadworks.net
config sip-gateway broadworks> set domain-alias 12.39.208.251
config sip-gateway broadworks> set domain-alias 12.39.208.252
config sip-gateway broadworks> set peer-identity
   sip:broadworks@companyABC.com
config sip-gateway broadworks> config server-pool
config server-pool> config server sip:12.39.208.251
config server sip:12.39.208.21> return
config server-pool> config server sip:12.39.208.252
config server sip:12.39.208.21> return
config server-pool> return
config sip-gateway broadworks> set user cxc
config sip-gateway broadworks> set password-tag abcXYX
config sip-gateway broadworks> return
config servers> return
```

For detailed on the SIP registrars and SIP registration, refer to the *Net-Net OS-E – Session Services Configuration Guide* and the *Net-Net OS-E – Objects and Properties Reference*.

H.323 servers

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. AA-SBC supports H.323-to-SIP, SIP-to-H.323, and H.323-to-H.323 calls.

For detailed information on the H.323 server configuration settings, refer to the *Net-Net OS-E – Objects and Properties Reference*.

SIP connections

A SIP connection configuration identifies the call admission point, or the PSTN lines from which SIP calls originate to the AA-SBC platform.

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 address-of-record (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’s transport information from the client registration.

DNS groups

A dns-group is a server configuration template for servers that don’t 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**.

Configuring directories

A directory is a database of user information—data such as name, group membership, address, position, office location, contact information, and any number of other identifiers. A directory is part of a centralized system that manages the SIP-addressable users who can participate in SIP sessions. SIP clients who wish to establish SIP sessions with other users will access the directory to look up user entries. If the lookup is successful, the AA-SBC platform then applies policies that control the actual SIP session.

Directories include:

- Active directory
- LDAP directory
- Notes directory
- Phantom directory
- Static directory
- XML directory
- Database directory
- CSV (comma-separated values)

Active directory

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 configuring the objects in the active-directory, 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 in your network.

CLI session

The following CLI session shows the `active-directory` object hierarchy.

```
NNOS-E> config
config> config vsp
config> config enterprise
config enterprise> config directories
config directories> config active-directory name
config active-directory name>
```

LDAP

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.

Configure the LDAP server on the AA-SBC platform if you don’t use a Windows server in your enterprise. This configuration sets how AA-SBC should recognize and query the LDAP 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*.

CLI session

The following CLI session shows the `ldap` object hierarchy.

```
NNOS-E> config
config> config vsp
config vsp> config enterprise
config enterprise> config directories
config directories> config ldap name
config ldap name>
```

Notes directory

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 configuring the objects in the notes-directory, you are providing AA-SBC with access to the Domino Enterprise Directory service. AA-SBC can then access the required databases to derive the recognized SIP addresses in your network.

CLI session

The following CLI session shows the `notes-directory` object hierarchy.

```
NNOS-E> config
config> config vsp
config vsp> config enterprise
config enterprise> config directories
config directories> config notes-directory name
config notes-directory name>
```

Phantom directory

The phantom directory is an internal directory that is derived from another directory. The phantom directory allows you to create and use multiple SIP aliases for each user. You do this by creating a phantom directory where additional SIP information is stored for the user.

Note: When you use the CLI to configure the `phantom` directory object, you are also required to specify the `tag, domain, parent-directory,` and `sip-address` properties for this object.

CLI session

The following CLI session shows the `phantom` object hierarchy, using a sample phantom directory named *“bw”*.

```
NNOS-E> config
config> config vsp
```

```
config vsp> config enterprise
config enterprise> config directories
config directories> config phantom bw
config phantom phantom1> set tag bw
config phantom phantom1> set domain companyABC.com
config phantom phantom1> set parent-directory vsp\enterprise\
   directories\active-directory east
config phantom phantom1> config sip-address
config phantom phantom1> set value %@uid%
config phantom phantom1> return
```

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).

CLI session

The following CLI session shows the `static` object hierarchy.

```
NNOS-E> config
config> config vsp
config> config enterprise
config enterprise> config directories
config directories> config static-directory name
config static-directory name>
```

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.

CLI session

The following CLI session shows the `xml-directory` object hierarchy.

```
NNOS-E> config
config> config vsp
config> config enterprise
config enterprise> config directories
config directories> config xml-directory name
config xml-directory name>
```

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 users listed in a database table instead of registered in an LDAP directory.

Note: When you use the CLI to configure the `database-directory` object, you are also required to specify the `tag, domain, and column` properties for this object.

CLI session

The following CLI session shows the `database-directory` object hierarchy for a database-directory named *abc*.

```
NNOS-E> config
config> config vsp
config> config enterprise
config enterprise> config directories
config directories> config database-directory abc
config database-directory abc> set domain companyABC.com
config database-directory abc> set tag fromNotes
config database-directory abc> set column name 1
```

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.

Note: When you use the CLI to configure the `csv-directory` object, you are also required to specify the `tag, domain, and source` properties for this object.

CLI session

The following CLI session shows the `csv-directory` object hierarchy for a csv-directory named *xyz*.

```
NNOS-E> config
config> config vsp
config> config enterprise
config enterprise> config directories
config directories> config csv-directory xyz
config csv-directory xyz> set domain companyABC.com
config csv-directory xyz> set tag fromNotes
config csv-directory xyz> set source ftp://myftp.companyABC.com/
users.xml
```

Configuring federations

Federations, as described earlier in this chapter, are groupings of servers that allow users of those servers to communicate, providing the ability to connect across organizations using the same or different SIP server deployments. The organizations, by joining the federation, designate each other as trusted federated partners.

Federations allow SIP users in different domains to communicate, where AA-SBC acts as the intermediate device, such as an access proxy for LCS, or a SIP connector for IBM Sametime. Federations are also needed where differing SIP server deployments need to interoperate, such as Microsoft LCS users conducting SIP sessions with IBM Sametime users.

The federation configuration object allows you to specify the name of the federation 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 already exists, the system opens that instance.

The following CLI session creates a federation named *LCS-IBM*, and includes the servers named LCS east and Sametime west

```
NNOS-E> config vsp
config vsp> config enterprise
config enterprise> config federations
config federations> config federation LCS-IBM
config federation LCS-IBM> set server vsp\enterprise\servers\lcs east
config federation LCS-IBM> set server vsp\enterprise\servers\sametime
 west
```

Configuring 3PCC servers

Third party call control (3PCC) 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:

- 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 clients
- Redirect calls to any other desktop or mobile telephone.

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

For detailed information on the 3PCC configuration settings, refer to the *Net-Net OS-E – Objects and Properties Reference*.