

**Oracle® Communications Application
Session Controller**

Session Services Configuration Guide

Release 3.7.0

May 2016

Copyright ©2016, 2005, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

Preface	xi
About Net-Net OS-E® Documentation.....	xi
Revision History	xii
Conventions Used in This Manual	xiii
Typographical Conventions.....	xiii
Acronyms	xiii
How Net-Net OS-E Operates on SIP Sessions.....	19
About This Chapter	19
About SIP Sessions	19
How the Net-Net OS-E Establishes SIP Sessions.....	20
SIP Session Processing Hierarchy	21
Normalization	23
Creating the Pre-Session Configuration	24
Blocking SIP Methods.....	24
Operating on SIP Headers.....	25
Creating the Default Session Configuration.....	26
Creating the Session Configuration	27
Session Policy Entries — Rules, Condition Lists, and Session Actions.....	28
Creating Session Configuration Pools	28
Net-Net OS-E Policy Overview	29
Creating SIP Session Policies	29
CLI Session.....	29
Policy Components and Ordering	30

Creating the Rules, Conditions, and Actions	31
Policy Rules	31
Policy Conditions	32
Policy Actions in the Session Configuration	34
Creating Policies Using the CLI	36
Denial of Service Policies	39
About This Chapter	39
Denial of Service Prevention Overview	39
DOS Policies	40
How DOS Policy Object Properties Work Together	41
How the Inactivity Timer Works	43
Using Operators and Regular Expressions	43
Setting DOS Security Levels in the Net-Net OS-E Management System	44
Low security — Set DOS Security to “Low”	45
Medium Security — Set DOS Security to “Medium”	46
High Security — Set DOS Security to “High”	46
Sample DOS Configuration	47
Configuring DOS Policies in the CLI	47
Examining the DOS Packet History	49
Administering the DOS Database	50
Managing DOS Policy Results	50
Configuring Dial Plans	51
About This Chapter	51
Dial Plan Overview	51
Configuring Dial-Plan Routes	52
Inbound and Outbound Call Normalization	53
Operating On the SIP FROM Header	54
Configuring the Dial-Plan Source Route	55
Configuring Dial-Plan Routing Arbitration	56
Configuring a Dial Route For the Request URI	57
Configuring Dial-Plan Normalization	58

Configuring the Dial-Prefix	58
Viewing the Call Routing Tables	59
Dial-plan Related Show Commands.....	60
Configuring Location Services	61
About This Chapter	61
How the Net-Net OS-E Stores Location Information	61
Address-Of-Record Static Bindings	62
Tags Associated With an Address Of Record.....	62
Configuring the Location-Service	63
Configuring SIP Registration Services.....	65
About This Chapter	65
About SIP Registration and Registration Plans	65
Enabling the Local Registration Service.....	67
Configuring Registration-Plan Settings	68
Delegating Registrations	68
Authenticating With the Upstream Registrars.....	71
Pinging Upstream Registrars For Availability	72
Creating Registration-Plan Routes	73
Proxying Registrations	74
Sample Proxy Registration Configuration.....	74
Client Authentication Over Proxy Registration	75
Address-Of-Record Bindings.....	76
Proxy Registration Network Example	77
Performing Other Actions On SIP Registrations	79
Forwarding Registrations.....	79
Redirecting Registrations	80
Tunneling Registrations.....	81
Discarding Registrations.....	81
Blocking Registrations	82
Registration Throttling	82
Controlling Registration Admission Per VSP.....	83

Registration Multicasting	84
Registration Normalization	85
Viewing the Registration Routing Table.....	86
Configuring a PSTN Gateway	87
Session Recording and Monitoring	89
About This Chapter	89
Policy-Based Media Recording and Playback.....	89
Enabling Media Anchoring and Recording	91
Configuring Call Monitor Groups and Endpoints	92
Playing Back Recorded Calls	92
Recording File Transfers and IM Sessions.....	93
For Tile Transfer Recording.....	93
For Recording IM Sessions	93
Storing Recorded Data In Specific Locations	94
Simultaneous Recording To Multiple Data Locations	95
Setting Up a Pre-Recorded Call Announcement.....	96
Managing Pre-Recorded Announcements For Possible Failover.....	98
Configuring Routing Arbitration	101
About This Chapter	101
What Is Routing Arbitration?	101
About Carriers and Gateways	102
Terminology	103
How Routing Arbitration Works	103
Sample Carrier Network.....	104
Configuration Steps	105
Configuring the Enterprise Servers (Sip Gateways).....	105
Configuring the Server-Pools	107
Configuring SIP Connections	108
Configuring Carriers and Switches.....	109
Configuring Trunk Groups	110
Configuring Hunt-Groups	111

Configuring the Dial-Plan Arbiter	112
Configuring Secure Trunking Networks	115
About This Chapter	115
Sample Secure Trunking Networks	116
Call Traversal In the Secure Trunk	117
Configuration Steps	117
Net-Net OS-E Encryption/Decryption Policies.....	118
How the OS-E Performs Encryption and Decryption.....	118
Applying Encryption and Decryption Policies	119
Configuring Route Server Services	121
About This Chapter	121
Downloading and Installing the Route Server Import Client.....	123
Before Installing the Route Server Import Client	123
Downloading and Installing the Route Server Import Tool.....	126
HTTPS Support for Call Rate Files Transferring	133
Using Variables in the Route Server Import Tool.....	137
Using Different Rate Sheet Formats.....	137
Configuring the Route Server.....	139
Enabling Route Server on the OS-E master.....	139
Configuring Diameter Servers	140
Defining Diameter Groups Under the VSP Configuration.....	140
Configuring the Route Server Service	141
Configuring Enterprise Servers and Carriers.....	142
Configuring Routing Arbitration	142
Using the Import Client Features.....	143
Import Rates	143
Import LATA.....	150
Import Region Code	151
Backing Up Rates.....	152
Restoring Rates.....	153
Purging From the Route Server.....	153

Configuring DID Mapping	157
Importing DIDs.....	158
Retrieving Routes from the Route Server.....	171
Testing DID Ranges and Prefix Changes	175
Viewing the Audit Log.....	181
Displaying Route Server Version Information.....	182
Route Server Actions	182
Route Server Status	184
Performing Route Server Queries	185
Viewing This Document.....	191
Admission Control	193
About This Chapter	193
Call Admission Control.....	193
VSP Control.....	194
Server Control	196
User-Agent Control.....	197
Using the Session-Config Override	198
Registration Admission Control.....	198
VSP Control.....	199
Server Control	201
TLS Admission Control.....	201
VSP Control.....	202
Calling Groups	203
About This Chapter	203
Calling Groups Overview.....	203
Configuring Calling Groups	205
Net-Net OS-E SIP Trunking	207
About This Chapter	207
Sample SIP Trunking Network	208
Configuring a SIP Trunk.....	208

Configure the Enterprise SIP Gateway.....	209
Configure the Carrier Network.....	209
Create the Hunt-Group	210
Configure the Enterprise Dial-Plan	210
Configure the Carrier Dial-Plan	211
WebRTC Overview	213
What is WebRTC?.....	213
WebRTC Media Handling.....	214
Configuring WebRTC Using SIP Signaling Over WebSockets.....	229
Configuring WebSocket Listener Sockets	229
WebRTC Using OS-E REST Call Control APIs.....	232
Configuring the Multimedia Streaming Server	241
About This Chapter	241
Configuring MSS	241
Types of Supported MSS Calls	244
About This Appendix	251
Mean Opinion Score Overview	251
MOS Call Quality Statistics Gathering	252
Formulating MOS Results	252
Displaying MOS Results With the Net-Net OS-E Management System	254
About This Appendix	257
Call Detail Record Overview	257
SIP Call Legs	258
Displaying the CDR	259
CDR Field Descriptions and Data Types	261
Sending CDRs to External Databases	267
About This Appendix	269
DTMF Overview	269
Signaling Overview	270
RFC-2833 Overview	273
Audio Overview	274

DTMF Features	274
SIP Configuration Example	277
H.323/SIP Configuration Example	277
DTMF Troubleshooting	278
Troubleshooting Conversion From RFC-2833	278
Troubleshooting Conversion to RFC-2833 or Audio	284

Preface

About Net-Net OS-E® Documentation

The Net-Net OS-E references in this documentation apply to the Net-Net OS-E operating system software that is used for the following Oracle and third-party SBC products.

- Oracle Communications Application Session Controller (ASC)
- Oracle Communications WebRTC Session Controller (WSC)
- Oracle Communications OS-E Session Directory (SD) Session Border Controller (SBC)
- Oracle Communications 2600 Session Directory (SD) Session Border Controller (SBC)
- Third-party products that license and use Oracle Communications OS-E software on an OEM basis.

Unless otherwise stated, references to the Net-Net OS-E in this document apply to all of the Oracle and third-party vendor products that use Net-Net OS-E software.

The following documentation set supports the current release of the OS-E software.

- *Oracle Communications Application Session Controller System and Installation Commissioning Guide*
- *Oracle Communications Application Session Controller System and Installation Commissioning Guide Release 3.7.0M4*
- *Oracle Communications Application Session Controller Management Tools*
- *Oracle Communications Application Session Controller System Administration Guide*

- *Oracle Communications Application Session Controller Session Services Configuration Guide*
- *Oracle Communications Application Session Controller Objects and Properties Reference*
- *Oracle Communications Application Session Controller System Operations and Troubleshooting*
- *Oracle Communications Application Session Controller Release Notes*
- *Oracle Communications Application Session Controller Single Number Reach Application Guide*
- *Oracle Communications Application Session Controller Web Services SOAP REST API*
- *Oracle Communications WebRTC Session Controller Installation Guide*

Revision History

This section contains a revision history for this document.

Date	Revision Number	Description
June 28, 2013	Rev. 1.00	<ul style="list-style-type: none">• GA release of OS-E 3.7.0 software.
January 15, 2016	Rev. 1.10	<ul style="list-style-type: none">• Updates Chapter 9, Configuring Route Server Services.
May 17, 2016	Rev. 1.11	<ul style="list-style-type: none">• Adds <i>Oracle Communications Application Session Controller System Installation and Commissioning Guide Release 3.7.0M4</i> to the 3.7.0 doc set.• Updates Chapter 9, Configuring Route Server Services.

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.	[<i>portNumber</i>]
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 <i>name</i>
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 <i>directory</i> object.

Acronyms

The OS-E manuals contain the following industry-standard and product-specific acronyms:

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

Chapter 1. How Net-Net OS-E Operates on SIP Sessions

About This Chapter

This chapter describes how the OS-E processes SIP sessions using the session configuration, session policies, and normalization. These tools, using a specific hierarchy, instruct the OS-E on how to match, process, normalize, and direct (route) SIP traffic to meet the requirements of your network.

About SIP Sessions

SIP, the Session Initiation Protocol, and defined by a collection of Requests for Comment (RFCs) managed by the Internet Engineering Task Force (IETF), is a text-based messaging protocol for initiating interactive communication sessions between users.

A SIP session is a network connection between two (or more) SIP clients, with each SIP client accessing a SIP application that allows one client to communicate with the other. A SIP proxy server usually resides between the clients, providing security, monitoring, and control of the session. A client can be a SIP phone (an IP phone that connects to the Internet), or a PC having access to a SIP application, such as Windows Messenger, an instant messaging (IM) application where IM participants communicate with each other from text windows on their computer desktop.

Once the clients have started a session through SIP INVITE messages, they are able to conduct the session through session-specific message exchange. An active session uses the Transport Layer Security protocol (TLS), TCP, UDP and their associated ports to maintain SIP sessions. These SIP sessions include calls using telephones carrying voice, video, and data calls, multimedia conferences, and streaming media services.

SIP INVITE messages handled by the SIP proxy, the device that resides between the clients. Users register their network addresses-of-record with a SIP registrar so that SIP callers, using their hosted SIP applications, can look up and contact other SIP users who also have their address information in the registrar's database.

The OS-E system, operating as SIP proxy and a registrar, gathers registration and address-of-record information to locate and connect SIP callers. Registration ensures that the call goes through the OS-E system so that policies can be applied and enforced on the SIP session. Registration information is shared between registration peers, where each peer provides updates to each other about registered SIP users, and to what address a call might be forwarded, if handled by a specific service provider.

How the Net-Net OS-E Establishes SIP Sessions

SIP sessions are usually instant messaging (IM) sessions, such as sessions between Windows Messenger or Office Communicator clients, or SIP voice calls between SIP phones using SIP PBX equipment. Video conferencing and other SIP host-based applications use SIP and related protocols to support a SIP session. Regardless of the actual SIP application, all SIP sessions use signaling and media streaming to open and pass SIP messages over a session.

When a SIP session request arrives at the OS-E system, and after the request passes the OS-E's pre-session configuration, the "from" and "to" servers are determined using the information contained in the SIP request. The SIP message is forwarded to the SIP protocol for processing (using the SIP protocol "stack") so that the SIP session can be opened. When the session opens, the OS-E combines all session configurations (default session configuration, policy rules and conditions, and so on) and applies the configurations at various points to all messages associated with the SIP session.

Session requests that initiate with the SIP INVITE method differ from session requests that initiate using other SIP methods, such as REGISTER, INFO, and OPTIONS. SIP INVITE messages open a SIP session called a *dialog*, where the session is held open for messages between the caller and the recipient, and is identified with a call session ID that is persistent throughout the call session. “To:” and “From:” tags are then assigned a string value to identify messages that are part of the same call session. The session remains open until a SIP call participant (sender or receiver) sends a BYE message to the other participant. In other words, this translates to a SIP call participant simply hanging up the SIP phone, for example, or an instant messaging (IM) user closing an IM window.

Other SIP sessions, such as SIP REGISTER sessions, are short-lived and are only held open for 30 seconds. These types of sessions are outside of the dialog session that is created using the SIP INVITE method.

Keep in mind that while the SIP session is open, the OS-E performs as the proxy between the SIP caller and the SIP call recipient. Both the caller and recipient are each accessing a SIP server, SIP PBX, or other host-based SIP application.

IP protocols, network services, and database services running on the OS-E proxy allow to you to capture the SIP call detail records for accounting applications, record, monitor and playback SIP sessions, and query the OS-E call database for denial-of-service activity that could interrupt call throughput and services.

RADIUS, SYSLOG, and DNS, for example, are other remote services that the OS-E uses to support SIP call sessions and session record detailing.

SIP Session Processing Hierarchy

The OS-E uses a processing hierarchy to manage inbound and outbound SIP sessions. This hierarchy places configuration objects at a specific precedence, where certain objects of the configuration are checked before others. When a session match occurs, meaning that inbound SIP traffic (using the SIP header, URL, or other criteria) matches some level of the processing hierarchy, the OS-E then takes that appropriate action, such as connecting a caller, normalizing the call and forwarding it to another SIP gateway or call endpoint, or even dropping or disconnecting a call.

The following number list identifies the processing sequence when the OS-E evaluates a SIP message.

1. pre-session-config
2. default-session-config
3. default-policy
4. server inbound session-config
5. server inbound normalization
6. dial-plan/reg-plan normalization
7. dial-plan/reg-plan->arbiter session-config
8. dial-plan/reg-plan->route normalization
9. dial-plan/reg-plan->route session-config
10. named session config received via LCR or the ASC external policy service via WSDL
11. outbound-policy
12. server outbound session-config
13. server outbound normalization
14. server outbound normalization session-config

Session configurations are "layered" on top of each other so the last layer takes precedence. For example, in the above hierarchy, the **server outbound normalization session-config** takes precedence above all others.

If a particular layer has nothing configured for an entry that is defined by a previous layer, then the previous layer settings remain intact. For example, if the default session-config has a "to-uri-specification" defined and the policy does not, then the default-session-config setting remains.

A session configuration is applied just before the message is sent to its destination. If you have a server inbound-session-config that modifies the domain of the request-uri and to-uri, and because this change does not take effect until the message is forwarded, it will not affect the dial-plan/reg-plan lookup. For example, an inbound SIP INVITE has the following request-uri:

```
INVITE sip:+121020@foo.com:5060;transport=tcp SIP/2.0
```

The server inbound-session-config has a "request-uri-specification" that modifies the domain to "bar.com". Because the session-config application happens after each session-config has been evaluated, the dial-plan/reg-plan lookup is done based on request-uri:

```
INVITE sip:+121020@foo.com:5060;transport=tcp SIP/2.0
```

Rather than,

```
INVITE sip:+121020@bar.com:5060;transport=tcp SIP/2.0
```

Normalization

Unlike the session configuration, normalization modifies the message at the time it is evaluated. Consequently, normalization can affect a dial-plan/reg-plan lookup or a future normalization match. For example, an inbound INVITE has the following request-uri:

```
INVITE sip:+121020@foo.com:5060;transport=tcp SIP/2.0
```

The server-pool inbound-normalization has a configuration that modifies the request-uri user to "5552220001". Because normalization happens immediately, the dial-plan/reg-plan lookup is done based on the following request-uri:

```
INVITE sip:5552220001@foo.com:5060;transport=tcp SIP/2.0
```

Rather than,

```
INVITE sip:+121020@foo.com:5060;transport=tcp SIP/2.0
```

Cascading normalization can result in multiple modifications to a given header. For example, an inbound INVITE has the following request-uri:

```
INVITE sip:2078541000@foo.com:5060;transport=tcp SIP/2.0
```

The server-pool inbound-normalization has a configuration that strips the '207' from the request-uri-user resulting in the following URI:

```
INVITE sip:8541000@foo.com:5060;transport=tcp SIP/2.0
```

The dial-plan/registration-plan lookup directs the call to a server that has an outbound normalization that appends '543' to the request-uri-user which is forwarded as:

```
INVITE sip:5438541000@foo.com:5060;transport=tcp SIP/2.0
```



Note: Normalization only affects user values, while a session configuration can manipulate many fields in a header. The OS-E can only change user values before any dial-plan/reg-plan, arbiter, or subsequent normalization match.

Creating the Pre-Session Configuration

The *pre-session configuration* describes how the OS-E should behave based on SIP message headers and enumeration settings, and if certain SIP methods should be blocked, preventing the SIP session from being established.

Before you create policies, you can edit the pre-session configuration to preemptively take action on a SIP session request before the OS-E initiates the session. This allows you to globally block or modify certain types of SIP sessions before a session is started.

The pre-session configuration supports the following operations:

- Blocking configured SIP methods
- Modifying SIP headers (including discarding SIP packets)
- Modifying SIP ENUM settings
- Setting a directive for unregistered SIP senders

Blocking SIP Methods

The SIP methods that you can block on a SIP session request include:

- INVITE
- ACK
- OPTIONS
- BYE
- CANCEL

- REGISTER
- MESSAGE
- INFO
- NOTIFY
- SUBSCRIBE
- REFER
- PRACK
- PUBLISH
- UPDATE
- PING

By default, the OS-E allows all SIP methods.

CLI Session

The follow CLI session blocks all SIP session requests containing the SIP INFO method.

```
config> config vsp pre-session-config
config pre-session-config> config block-method-settings
config block-method-settings> set admin enabled
config block-method-settings> set block method info
```

Operating on SIP Headers

The **sip-header-settings** object allows you to modify or alter SIP headers using configured rules and actions based on the header names and value settings.

CLI Session

The follow example CLI session prevents SIP packets from the user named *evilBadGuy*. The CLI session does the following:

- Enables the **sip-header-settings** administration state.
- Creates the user-defined SIP session rule named *dropPackets*.
- Creates an informational text field to help describe the rule.

- Sets the **discard-packet** action.
- Sets the condition that matches the name (value) in the SIP header (evilBadGuy) value.

```
config> config vsp pre-session-config
config pre-session-config> config sip-header-settings
config sip-header-settings> set admin enabled
config sip-header-settings> config rule dropPackets
Creating `rule dropPackets`>
config rule dropPackets> set description "Drop SIP requests
    from name evilBadGuy"
config rule dropPackets> set action discard-packet
config rule dropPackets> set condition match-value evilBadGuy
```

Creating the Default Session Configuration

Each OS-E system uses a *default session configuration* that contains the entire possible configuration that can be applied to a SIP session in the absence of a configured policy that matches the SIP session request. This is default preemptive behavior of the OS-E as an enterprise SIP proxy.

Unlike the **pre-session-config**, the **default-session-config** configuration operates on the SIP session once the session is started, but before the call is forwarded or connected, and in the absence of any OS-E policy that matches a SIP session request. Destination server policies, however, are applied to the session, if configured, along with the default session configuration. In cases where a policy match does exist, the OS-E uses the session configuration settings that you define under each policy.

One strategy that you can use is to open each object to display the OS-E-provided default settings. Some settings may be appropriate for your network, while others may not. The *Net-Net OS-E – Objects and Properties Reference* provides a complete description of each **default-session-config** object.

CLI Session

The following example CLI session shows a sample default session configuration that does the following:

- Sets the transport and port.
- Sets the directive to allow SIP traffic received on the OS-E proxy to the to the destination SIP server.

- Performs local authentication of the SIP user who has established the SIP session.
- Configures SIP call detail records to be forwarded to the specified RADIUS accounting group
- Sends an alert message and session information to the OS-E event log.
- Blocks the specified media types (audio and video) from entering the SIP session.

```
config> config vsp default-session-config
config default-session-config> config sip-settings
config sip-settings> set transport TCP
config sip-settings> set port auto-determine
config sip-settings> return

config default-session-config> config sip-directive
config sip-directive> set directive allow
config sip-directive> return

config default-session-config> config authentication
config authentication> set mode local enabled
config authentication> return

config default-session-config> config accounting
config accounting> set target radius "vsp\radius-group 1"
config accounting> return

config default-session-config> config log-alert
config log-alert> set message-logging enabled
config log-alert> return

config default-session-config> config media-type
config media-type> set blocked-media-types audio any
config media-type> set blocked-media-types video any
```

Creating the Session Configuration

The OS-E logically applies a session configuration when there is a matching SIP call request with a configured dial-plan, registration-plan, or policy. In the case of a non-matching request, the session is governed by the **default-session-config** or a **default-policy** session configuration under **vsp\policies\session-policies**.

The OS-E parses the configuration file for SIP request matches in the following order:

1. Locate a matching **dial-** or **registration-plan** in the configuration.

2. If a **policy** is configured for the dial- or registration-plan, match the SIP request to the policy and apply the associated session configuration.
3. If a **session-config** is configured for the dial- or registration plan, apply the session configuration
4. If a **session-config-pool** is configured for the dial- or registration-plan, apply session configuration
5. If there is a matching **default policy**, apply the session configuration for the matching **rule**.
6. If there are no matches above, apply the **default-session-config** to the call session.

Session Policy Entries — Rules, Condition Lists, and Session Actions

Each OS-E session policy that you define is a single policy entry. Each policy consists of a series of *rules*, with each rule having a unique *condition-list* and *session-configuration*. The condition list describes the specific conditions that are evaluated based on configured criteria, such To: and From: URI and server conditions, date and time, user group memberships, and from the SIP message itself.

Using policies, rules, and conditions lists, the OS-E system decides how to handle or act on the SIP traffic using the *session configuration*. The session configuration is the same as the default-session configuration. However, the session configuration is dedicated to the overlaying policy rule, governing only those SIP sessions that match the rule and the condition list. The policy session configuration always overrides the default session configuration when a policy match occurs with the SIP session.

If the condition list evaluates to TRUE (i.e., matched), then the session configuration is applied to the rule. Additionally, the conditions in the condition list can be AND'd and OR'd together to make a policy decision.

All policy rules within a policy are applied one after another so that the most specific policies are applied first (User, Group, Server, and Default).

Creating Session Configuration Pools

The session configuration pool is a mechanism for creating a session configuration that can be referenced through a dial plan. By creating a specific session configuration using a unique name, you can re-use it for all applicable dial plans without having to create multiple, identical session configurations.

Each session configuration in the pool uses a unique name that you specify. The configuration objects available within a **session-config-pool** entry are the same session objects available for the default or pre-session configuration.

For information on referencing a session configuration from the pool, refer to Chapter 3, “Configuring Dial Plans.”

Net-Net OS-E Policy Overview

Polices allow you to control and monitor the SIP sessions that traverse the enterprise network. Policies use rules, conditions, and session actions that are executed in a logical order whenever a SIP client establishes a session with a SIP recipient. When the OS-E evaluates the SIP call request, and if the call session is allowed based on matched criteria, specific actions are applied at various points during the call session. A condition list determines the OS-E actions that are applied to a SIP call session.

Creating SIP Session Policies

You can create SIP session policies using the command line interface (CLI). A session policy can be as simple or as complex as you can make it. Policies that block certain types of traffic can be relatively simple, while policies containing multiple rules and conditions, as well as those that you build with regular expressions, are more difficult to create. Starting off with some simple, or less-complex session policies can serve as building blocks for more detailed policies.

CLI Session

The following CLI session opens the **policy** configuration object on the OS-E system with the user-specified name called *companySierra*, a rule called *blockSession*, and a text description of the policy to be created. Also shown are the two objects that define the session policy, the **condition-list** and the **session-config**.

```
config> config vsp policies
config policies> config session-policies
config session-policies> config policy companySierra
Creating 'policy companySierra'
config policy companySierra> config rule blockSession
Creating 'rule blockSession'
```

```

config rule blockSession> set admin enabled
config rule blockSession> set description "Rule to block sessions
    based on matched conditions."
config rule blockSession> config condition-list
config condition-list> return
config rule blockSession> config session-config

```

Note that the **session-config** object contains the same subobjects as the **default-session-config** object, as described in the previous section. The difference is that the **session-config** applies to those SIP sessions where an evaluation of the session determines one or more policy matches.

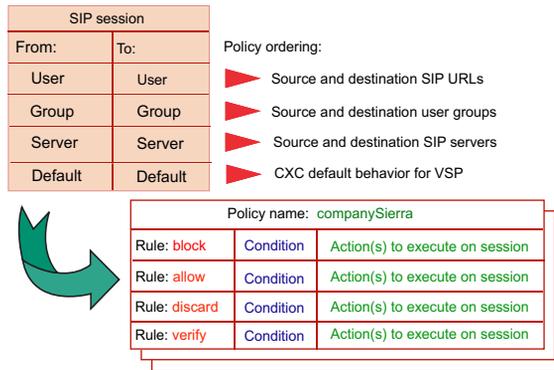
Policy Components and Ordering

A policy is a rule that you create using a unique condition-list and a session configuration. The condition list defines the evaluation criteria that the OS-E uses to match SIP session with configured policies. The session configuration then prescribes the action to perform on the session if a match occurs. The following image illustrates the policy component structure on the OS-E system.

Policy name: <i>companySierra</i>		
Rule: <i>block</i>	Condition	Action(s) to execute on session
Rule: <i>allow</i>	Condition	Action(s) to execute on session
Rule: <i>discard</i>	Condition	Action(s) to execute on session
Rule: <i>verify</i>	Condition	Action(s) to execute on session

The following image illustrates the relationship and ordering of the session qualifiers (User, Group, Server, Default) that the OS-E uses to decide which policies apply to the SIP session. Policies on a session are ordered so that the most specific policies are applied first in the order **User->Group->Server->Default**. When the OS-E applies policies of the same level to a session, the **From** user policies have a higher priority than the **To** user.

Qualifiers that decide policies to apply to the SIP session



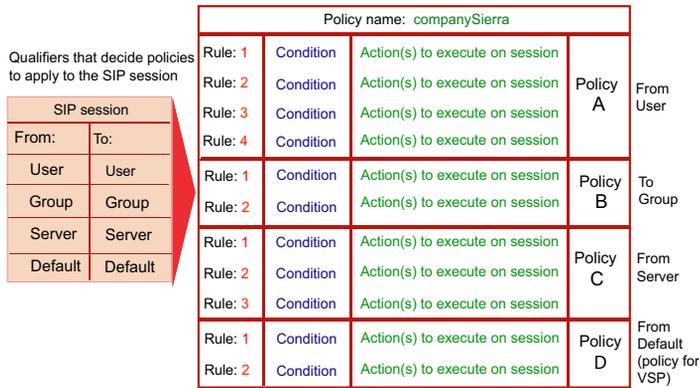
When there are multiple policies for the both the direction (From, To) and the level (User, Group, Server, Default), policies apply in the order that they appear in the policy configuration tree.

Creating the Rules, Conditions, and Actions

Policy Rules

The OS-E extracts rules from policies before applying the rules to the active SIP session. Rules within a policy are placed on the session in the order in which they are specified in the policy. Once configured, you can adjust the order of the rules within policy so that some rules maintain a higher precedence than others.

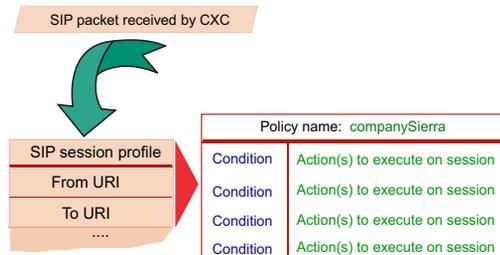
The image below illustrates how the SIP session is using Policy A through Policy D on a SIP session based on policy matching. Policy A maintains the highest priority and executes Rules 1 through Rule 4. Policy B, Policy C, and Policy D follow in sequence.



Policy Conditions

When SIP packets arrive at the OS-E for session establishment, and if the **pre-** and **default-session-config** does not block the SIP packet processing, the OS-E proceeds to build a session profile. The session profile contains data elements that are used to match the conditions that are contained in each rule. This means that the condition must match a field in the profile if an action is to take place on the SIP session.

The following image illustrates how the OS-E builds a session profile using fields in the SIP packet for condition matching and policy execution on the session.



The session profile under **vsp\policies\session-policies\policy\rule\condition-list** can be configured to match the following conditions associated with the received SIP packet:

sip-message-condition

- **Match properties:** box, to-uid, from-uid, ip-interface, direction, transport, remote-ip, local-ip, remote-port, local-port, call-leg, message-type, request-method, response-code, header, content, result-code, result-description, request-uri-user-host, to-user-host, from-user-host, cseq-method-type, request-uri, response-string, call-id, from, to, cseq, private-remote-ip, private-remote-port, uac-public-ip, uac-public-port uac-public-transport, pushing-thru-proxy, media-types, public-local-ip, content-type, user-agent

Example:

```
config condition-list> set sip-message-condition from match spam.com
```

from-uri-condition, to-uri-condition

- **Match properties:** scheme, identifier, user, host, port, ttl, method, url, other, transport

Example:

```
config condition-list> set from-uri-condition user match  
evilBadGuy@spam.com
```

request-uri-condition

- **Match properties:** scheme, identifier, user, host, port, ttl, method, url, other, transport

Example:

```
config condition-list> set request-uri-condition user match  
evilBadGuy@spam.com
```

from-server-condition

- **Match properties:** tag, address, server-type

Example:

```
config condition-list> set from-server-condition address match  
210.46.5.1
```

date-time-condition

- **Match properties:** day, month, hour, minute, date, year

Example:

```
config condition-list> set date-time-condition month match June
```

user-group-condition

- **Match properties:** User group associations found in directories. (to-user-group, from-user-group)

Example:

```
config condition-list> set user-group-condition to-user-group match
companySierra
```

action-condition

- **Match properties:** (none, call-control, presence-subscribe, presence-end-subscription)

Example:

```
config condition-list> set action-condition call-control
```

Policy Actions in the Session Configuration

The policy session-configuration describes the actions to perform on the call session based on matching conditions. As described earlier, SIP sessions for which there are no matching conditions will use the default-session-config to control the VSP and how it performs actions on the session.

Each policy action has a precedence associated with it. The precedence determines the point in the session where that action is applied. Some actions operate directly on the SIP packet where the “From:” and “To:” URI may be altered. Other actions enable signaling and media services on the session.

The following table lists some of the policy actions and the logical order in which they are processed on the SIP session

Session action	Description
sip-settings	Directs the session to a SIP server (IP address) over the configured transport (UPD, TCP, or TLS).
sip-directive	Provides instructions for the SIP session.ocket. If set to DISCARD or REFUSE, no additional processing is performed on the session.

Session action	Description
to-uri-specification	Alters the To: field of the SIP message so that the session is redirected to the specified URI.
from-uri-specification	Alters the From: field of the SIP message so that the SIP session appears to have originated by the specified URI.
request-uri-specification	Specifies where each field of the outgoing request-uri will be derived from.
contact-uri-settings-in-leg	Specifies where the OS-E derives the content of the CONTACT header from when it forwards a message to a UAC.
contact-uri-settings-out-leg	Specifies where the OS-E derives the content of the CONTACT header from when it forwards a message to a UAS.
presence	Enables presence translation services, such as LCS -to-LCS, and IBM Sametime-to-LCS, and presence value mappings (busy, online, be-right-back, etc.)
registration	Enables URI registration caching to the OS-E local registration database.
authentication	Configures the authentication method to use on this session, such as none, accept, reject, local, RADIUS, DIAMETER, and directory.
accounting	Specifies the accounting target, such a RADIUS accounting group, to which session accounting information is sent.
media	Configures media anchoring, NAT traversal, packet marking, and encryption on SIP media-type sessions, such as SIP INVITE sessions.
in-encryption	Specifies encryption parameters for inbound calls.
out-encryption	Specifies encryption parameters to outbound endpoints.
media-type	Sets the media types that are allowed and prohibited during the session.
log-alert	Generates an alert to the OS-E event log if the session is created.
refer-settings	Enables or disables call parking compatibility settings for the Sylanro SIP for Business initiative.
instant-messaging	Configures instant messaging (IM) directives, as well as alert settings that are forwarded to the configured logging target.
instant-messaging-content	Applies word and URL filtering lists to the IM session. The IM session is checked for words and URLs contained in the lists. Word and URL matches are struck from the IM session.

Session action	Description
file-transfer	Configures the settings that allow file transfers over the SIP session, such as anchoring, recording, and virus scanning.
forking-settings	Configures simultaneous ringing of SIP destination phones.
header-settings	Configures SIP headers to strip from the SIP messages before forwarding the call to the destination. Vendor-specific modes.
trusted-interface-settings	Provides an interface to allow non-LCS devices to interact with LCS clients.
session-control-settings	Specifies whether the OS-E should process policy on only the first or on all messages in a session.
playback-call-settings	Enables playback of the last recorded SIP call in a specific To/From pair.
csta-settings	Enables a CSTA session, allowing VoIP call control features over SIP sessions, such as instant messaging.
sip-session-timers-settings	Sets the values of SIP session timers.
egress-routing-setting	Configures a geolocation to match against configured IP interfaces.
third-party-call-control	Specifies the WAV files that the OS-E should play when it is acting as a third-party call controller.
uui-header	Specifies the UUI header that can be used for passing the universal call ID (UCID) and other session information to the NICE media server.
3GPP	Specifies 3rd Generation Partnership Project (3GPP) systems.
response-translation-settings	Maps a new status code and, optionally, phrase to a received code.
accounting-data	Adds a custom data field to the accounting record.
codec-specific-parameters	Adds an a=fmtp line to the SDP.

Creating Policies Using the CLI

This section covers a simple policy configuration using the CLI. The complexity of the policy determines which options you will need to configure in the condition list, as well as the specific actions to execute should a policy match occur in the SIP session.

The following CLI session

- Creates the policy named *companySierra*.

- Creates, enables, and configures the rule named *blockSession* with a text description of the rule.
- Opens the **condition-list** object and sets the SIP message conditions so that the OS-E examines SIP INVITE requests from the user *sales* at *sip:sales@spam.com* to the user named *bob.smith@companySierra.com*.
- Opens the **session-config** object and sets a SIP directive to discard all SIP INVITE traffic from *Sales* to *Bob Smith*.
- Sends an alert message to the configured syslog server.

The SIP message in this example is as follows:

```
INVITE sip:bob.smith@companySierra.com
Via: SIP/2.0/UDP server.spam.com:5060
From: Sales <sip:sales@spam.com>
To: Bob Smith <sip:bob.smith@companySierra.com>
Call-ID: 1234@server.spam.com
Cseq: 1 INVITE
Contact: Sales <sip:sales@server.spam.com>
Content-Type: application/sdp
Content-Length: 154
```

CLI Session

```
config> config vsp policies
config policies> config session-policies
config session-policies> config policy companySierra
config policy companySierra> config rule blockSession
config rule blockSession> set admin enabled
config rule blockSession> set description "Rule to block INVITE
sessions based on matches from certain sites."

config rule blockSession> config condition-list
.list> set operation AND
.list> set mode evaluate
.list> set sip-message-condition request-method match INVITE
.list> set from-uri-condition url match sip:sales@spam.com
.list> set from-uri-condition display-name Sales
.list> set to-uri-condition url match sip:bob.smith@companySierra.com
.list> set to-uri-condition display-name "Bob Smith"
.list> return
config rule blockSession>

config rule blockSession> config session-config
config session-config> config sip-directive
config sip-directive> set directive discard
```

```
config sip-directive> return  
config session-config> config log-alert  
config log-alert> set message-logging enabled
```

Chapter 2. Denial of Service Policies

About This Chapter

The chapter describes how you can perform Denial of Service (DOS) queries to the OS-E database, and how to create policies that prevent DOS attacks to the OS-E system.

Denial of Service Prevention Overview

A Denial of service (DOS) attacks is usually a flood of meaningless network traffic from a sender who intends to disrupt or totally disable services at a network destination. The OS-E system provides transport-layer, SIP-message, and URL policy definitions to detect DOS attacks. Queries allow you to sort and view incoming and outgoing traffic to better define policies. Policies determine if a packet is attacking the OS-E, and if so, the configured action is applied to that traffic. These tools quickly identify and shutout useless traffic, limiting any damage that might be caused by DOS attacks.

The OS-E uses the integrated database to record all packets that are transmitted or received by the system. The records are stored in specific tables that the OS-E can then access for queries and for policy execution. Activities through the transport layer, such as file transfers and SNMP walks, are stored in the transport layer table. The transport engine accesses that table for transport-level queries and policies. For each entry, the table records the following information from the TCP header:

- Remote IP address
- Remote port
- Local port

- Protocol

The SIP table contains entries for all SIP-related activities. The table includes all TCP-header information, as well as fields in the SIP header.

DOS Policies

The DOS transport, SIP, and URL policies use condition list setting to determine the point at which activity is considered to be part of a DOS attack and what action is to be taken. The action taken by the OS-E depends on whether the attack was identified by the transport, SIP, or URL policy. A policy fires at the frequency defined by the **period** property, scanning the database over the course of the last period looking for matches to the policy.

Transport Policy Operations

Transport policies are based on the TCP, UDP, and IP headers. The OS-E keeps a count of each time a policy match occurs. When the count exceeds the allowable threshold set in the condition list, the OS-E creates a dynamic rule, based on the criteria that resulted in a DOS attack declaration, in the kernel filter. Packets matching the pattern defined in the filter are dropped. The kernel rule keeps a count of dropped packets for later comparison.

The advantage to transport filters is that packets are dropped as soon as they enter the box, requiring very little processing. The disadvantage is that the filter is only set based on TCP header fields, which limits the flexibility. For more detailed policy, you should set SIP policy.

SIP Policy Operations

The SIP policy operates similarly to transport policy, but operates on the SIP header. The OS-E maintains a record of each hit to the policy, and the statements of the condition list define when that data constitutes an attack. Unlike the transport filter, however, the SIP policy comparison is executed in the application layer of the OS-E software. With the SIP policy, you configure the action to take as a result of exceeding the threshold settings. Because you have many more fields to include as part of your filter, you have greater granularity in the filter design. The upper-layer processing, however, requires more CPU cycles.

URL Policy Operations

The URL policy operates on URL regular expressions. All action is taken based solely on the URL. When a URL meets the criteria defined, the implicit action is to drop the packet(s). The URL policy detects when the same URL has gone through the OS-E a specified number of times over a specified number of seconds. When the OS-E detects excessive appearances of a URL (for example, someone SPIMing your network with an ad for their Web site or a virus that self-propagates via links in IMs), it blocks future IMs containing that same URL, regardless of who the IMs appear to come from.

How DOS Policy Object Properties Work Together

When configuring a DOS policy, you:

1. Define the filter criteria with a condition list
2. Select the columns that you want to observe with a select statement.

The sort criteria are applied to the entire message database. Using one or more set condition statements, you define which packets will be considered further. For example, if your condition statement sets remote-ip to match 1.2.3.4. and sets the local-port equal to radius, all packets that originated from IP address 1.2.3.4 and are destined for the RADIUS port, are copied into a new table.

Select the columns from the header fields from which you want your final result set built from. So from the table created above, you may want to compare the local-ip and remote-ip to determine the count of potentially questionable packets. It is the result of this final compare that is measured against the threshold you set to determine the next action.

The following example is an aggregation created at the transport level from selecting remote-ip, remote-port, and protocol. (DOS SIP policy works in the same manner.)

The resulting table looks as follows:

Timestamp	remote-ip	remote-port	protocol
04:44:00	1.2.3.4	100	UDP
04:44:06	10.10.10.10	200	UDP
04:44:13	1.2.3.4	100	UDP
04:44:18	1.2.3.4	150	UDP
04:44:22	1.2.3.4	100	UDP
04:44:27	1.2.3.4	100	UDP
04:44:35	1.2.3.4	100	UDP

Additionally, this sample transport policy has the following properties:

- threshold=4
- period=30

From the table, the system keeps a count of instances of remote-ip/remote-port/protocol occurrences for each time period. The count for the table above would look as follows:

Occurrence	Count	Notes
1.2.3.4/100/UDP	4	The fifth occurrence fell in a new period. However, the threshold is four, so this constitutes a DOS attack!
1.2.3.4./150/UDP	1	The different port causes this packet to be counted separately.
10.10.10/200/UDP	1	None

If the message queue fills, regardless of whether the period interval has expired, the OS-E immediately executes all DOS policies.

How the Inactivity Timer Works

The inactivity timer is a mechanism for ensuring that when the system denies access to an offending sender, there is a time at which the prohibition expires. Once the sender has ceased tripping the kernel rule, the dynamic rules created in response to the configured threshold being crossed, the duration of the inactivity timer determines when the sender can resume communications.

Prior to each interval, the transport engine checks the kernel rule counter and caches the data. At the next interval, the system compares the current value to the cached value. If the counter has gone up, the system overwrites the previous cached value with the current counter. If the value is the same, which indicates that no new packets have been caught by the policy, then the system checks the inactivity time stamp. Once the inactivity timer times out, the kernel rule is deleted.

Using Operators and Regular Expressions

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

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

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

Go to one of the following Web sites for complete instructions on forming regular expressions:

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

Setting DOS Security Levels in the Net-Net OS-E Management System

You can assign security levels to DOS policies using the OS-E Management System Web. The security levels are:

- None — DOS policies are not configured.
- Low security
- Medium security
- High security



Caution: Changing the DOS security level removes all DOS policies from the configuration file.

To change the DOS security level, perform the following steps:

1. From the OS-E Management System, click on **VSP** from the menu tree.
2. Select **Set DOS security level**.
3. Set the security level to **None**, **Low**, **Medium**, or **High**.

All policies have an inactivity timeout that determines how long a DOS rule remains in effect once the DOS attack stops. DoS rules remain in effect for as long as the DoS attack persists. All the above policies are set to 300 seconds, except the URL policy which is set to 1,000,000 seconds.

Low security — Set DOS Security to “Low”

Transport policies

- **LowSecurity_remoteIP**—If an IP address is detected more than the 30,000 times over a 30-second period, then the transport DoS engine will block it.
- **LowSecurity_subnet**—If IP addresses within the same 255.255.255.0 subnet are detected more than 60,000 times over a 30 second period, then the transport DOS engine blocks the entire subnet. This policy excludes packets directed at the SNMP port.
- **LowSecurity_sip**—Same as LowSecurity_subnet, but only considers packets directed at the SIP port 5060, detected 4000 times over a 30 second period.
- **LowSecurity_sip_tls**—Same as LowSecurity_subnet, but only considers packets directed at SIP TLS port 5061, detected more than 600 times over a 30 second period.
- **LowSecurity_snmp**—Same as LowSecurity_subnet, but only considers packets directed at SNMP port 161, detected more than 4000 times over a 30 second period.
- **LowSecurity_https**—Same as LowSecurity_subnet, but only considers packets directed at HTTPS port 443, detected more than 200 times over a 30 second period.
- **LowSecurity_safetynet**—If IP addresses within the same 255.255.255.0 subnet are detected more than 50,000 times over a 5 second period, then the transport DoS engine blocks the entire subnet.

SIP Policies

- **LowSecurity_remoteIP_alert**—If the same IP address is detected more than 200 times over a 10 second period, an alert in the event log is generated. The packets are not blocked.
- **LowSecurity_fromUser**—If the same "From" user is detected more than 1000 times over a 10 second period, the SIP DoS engine blocks it.

Medium Security — Set DOS Security to “Medium”

Same policies as Low security, plus the following:

Transport Policies

- **MediumSecurity_remotePort**—If an IP address/remote port is detected more than 800 times over a 15-second period, the transport DoS engine blocks it. This policy excludes packets directed at the SNMP port.

SIP Policies

- **MediumSecurity_SPIM**—If the same "From" user is detected more than 200 times over a 10-second period with the SIP MESSAGE request method, the SIP DOS engine blocks it. This policy blocks IM spam.
- **MediumSecurity_SPIT**—If the same "From" user is detected more than three times over a 10-second period with the INVITE request method, the SIP DOS engine blocks it. This policy operates on autodial, telemarketing calls.
- **MediumSecurity_socket_timeout**—If the same remote IP address is detected more than 40 times over a 10 second period, and if the connection resulted in a TCP connection timeout, the SIP DOS engine blocks it. This prevents a DOS attack in which the attacker opens TCP sockets, but does not use them.

URL Policies

- **MediumSecurity**—If the same embedded URL within SIP IM messages is detected more than 20 times over 60 seconds, messages with that URL are dropped. This policy blocks the spread of downloaded viruses that generate IMs to propagate the virus to recipients configured in the address book

High Security — Set DOS Security to ‘High’

Same policies as Medium security, plus the following:

SIP Policies

- **HighSecurity_bad_headers**—If the same remote IP address is detected more than 50 times over a 10 second period with a "bad" SIP header, the SIP DOS engine will block it.

- **HighSecurity_policy_rejected**—If the same "From" user was rejected by the session policy configuration more than 500 times over a 10 second period, the SIP DoS engine will block it.

URL Policies

The HighSecurity URL policy is the same as the **MediumSecurity** URL policy

Sample DOS Configuration

This section provides sample CLI sessions that configure policies that capture DOS attacks. For detailed information on using the actual **set** commands to define DOS queries and policies, refer to the *Net-Net OS-E – Objects and Properties Reference*.

Configuring DOS Policies in the CLI

The VSP **dos-policies** object allows you to define the transport, SIP, and URL policy condition lists, which define the point at which activity is determined to be part of a DOS attack and what action is to be taken.

Transport Policy

The properties you set in the transport policy object define the “rules” for applying the condition-list to the transport table. At the transport level, the OS-E can filter on data based on fields of the TCP header. These fields are **remote-port** and **protocol**.

The **transport-policy** specifies the following:

- The **remote-port** and **protocol** fields to examine (**select**).
- The frequency (**period** of time in seconds) between checks to the DOS database.
- The **threshold** on the number of matches that must be found in the database over the configured period of time before a DOS policy performs a filtering action.
- The **condition-list** property references the criteria for choosing which packets then get run through the **select** screening to build the final result set.

CLI Session

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

```
config policies> config dos-policies
config dos-policies> config transport-policy 1
config transport-policy 1> set description "Filter out bad guys"
config transport-policy 1> set admin enabled
config transport-policy 1> set select remote-port+protocol
config transport-policy 1> set threshold 50
config transport-policy 1> set period 45
config transport-policy 1> set condition-list "vsp policies
dos-policies transport-condition-list 1"
Creating 'vsp\policies\dos-policies\transport-condition-list 1'
config transport-policy 1> return

config dos-policies> config transport-condition-list 1
config transport-condition-list 1> set operation OR
config transport-condition-list 1> set condition remote-ip 10.10.10.10
config transport-condition-list 1> set condition remote-port 2525
```

SIP Policy

The properties in the **sip-policy** object define the “rules” for applying the **condition-list** to the SIP table. The SIP policy specifies the following:

- The TCP/UDP/IP/SIP header fields to examine (**select** property).
- The frequency (**period** of time in seconds) between checks to the DOS database.
- The **threshold** on the number of matches that must be found in the database over the configured period of time before a DOS policy performs a filtering action.
- The **action** to take on closing “bad” calls; filter or alert.
 - **Filter** discards all future calls that match the policy
 - **Alert** allows the packets to pass, but generates a log event.
- The **inactivity-period** removes a filtering action due to inactivity if the configured time setting expires.
- The **condition-list** reference adds the criteria for choosing which packets then get run through the **select** screening to build the final result set.

CLI Session

```
NNOS-E> config vsp policies
config policies> config dos-policies
config dos-policies> config sip-policy 1
config sip-policy 1> set description "Filter out INVITES from 1.2.3.4"
config sip-policy 1> set admin enabled
config sip-policy 1> set select from-user
```

```
config sip-policy 1> set threshold 50
config sip-policy 1> set period 45
config sip-policy 1> set inactivity-period 180
config sip-policy 1> set condition-list "vsp policies dos-policies
  sip-condition-list 1"
config sip-policy 1> return

config dos-policies> config sip-condition-list 1
config sip-condition-list 1> set operation AND
config sip-condition-list 1> set condition from-user exclude hal
config sip-condition-list 1> set condition result match policy-discard
config sip-condition-list 1> set condition header match .*dave.*
```

URL Policy

The properties in the **url-policy** object define the “rules” for applying the condition list to the URL table. The URL policy specifies the following:

- The frequency (**period** of time in seconds) between checks to the DOS database.
- The **threshold** on the number of matches that must be found in the database over the configured period of time before a DOS policy performs a filtering action.
- The **condition-list** list property reference defines which URL entries to examine, or those URL entries not to examine.

CLI Session

```
NNOS-E> config vsp policies
config policies> config dos-policies
config dos-policies> config url-policy 1
config url-policy 1> set description "Filter out IM virus"
config url-policy 1> set admin enabled
config url-policy 1> set threshold 5
config url-policy 1> set period 45
config url-policy 1> set condition-list "vsp policies dos-policies
  url-condition-list 1"
config url-policy 1> return

config dos-policies> config url-condition-list 1
config url-condition-list 1> set url-condition match *buddy*
```

Examining the DOS Packet History

The OS-E Management System **Call Logs** tab allows you to view the packet history in the DOS database for which there are configured DOS policies.

Administering the DOS Database

The **master-services** database and dos-defense objects allows you to administer the DOS database on a host device. You can configure the time of day and interval when the database is purged of old entries.

For detail information on administering the DOS database, refer to the *Net-Net OS-E – System Administration Guide*.

Managing DOS Policy Results

There are several mechanisms for observing the effectiveness of your DOS policy configuration:

- The system generates an SNMP trap and a log message each time a DOS policy detects a DOS attack.
- DOS status providers:
 - **show dos-collection**
 - **show dos-database-entry**
 - **show dos-query-status**
 - **show dos-recent-sip-from-user**
 - **show dos-recent-sip-ip**
 - **show dos-recent-sip-port**
 - **show dos-recent-transport-ip**
 - **show dos-recent-transport-port**
 - **show dos-rules**
 - **show dos-sip-counters**
 - **show dos-sip-summary**
 - **show dos-transport-counters**
 - **show dos-transport-summary**
 - **show dos-url-counters**

Chapter 3. Configuring Dial Plans

About This Chapter

This chapter describes how you configure the dial plans, the mechanism that instructs the OS-E system on how to route SIP INVITE session requests.

Dial Plan Overview

The **dial-plan** configuration object instructs the OS-E on how to route SIP INVITE sessions, directing a SIP phone call to a particular gateway based on the dial prefix or domain suffix. You can also create a custom session configuration specific to the dial plan.

When the OS-E receives an INVITE, it extracts the USER portion of the SIP header. If all characters are digits, the OS-E alters the URL to “sip:xxxx@.*”. This makes the domain a wildcard match, meaning the phone number can be in any domain. However, the phone number must match the prefix specified by the **request-uri-match** property in the **route** configuration object. If all characters are not digits, the OS-E does a suffix match (and the domain remains unchanged).

Dial plans determine the entries that appear in the call routing table. If a server referenced in a dial plan becomes unavailable, the OS-E removes the entry from the call routing table. However, the entry remains in the configuration.

There are five configuration objects associated with a dial-plan, each covered in the sections that follow:

- route
- source-route

- arbiter
- normalization
- dial-prefix

Configuring Dial-Plan Routes

The dial-plan **route** object specifies the portion of the SIP REQUEST URI on which to match so that the call is routed to a particular server. If the call matches the type prefix or suffix specified by the entry, the OS-E applies the entry's session configuration to the SIP call.

A dial-plan applies a session configuration in the following ways:

1. If there is a session configuration added specifically for the dial-plan, the OS-E uses the settings in the **vsp\dial-plan\route\session-config** path. This is a custom session configuration pool entry that you configure as a sub-object of the dial-plan route.
2. If there is a reference to session configuration that is part of a pool, then the OS-E uses the settings in the referenced object, as configured in the VSP **session-config-pool** object.
3. Otherwise, no session configuration applies to the dial-plan.

The following CLI session creates three dial plan route, each containing a specific type of URI match on the SIP call request. Each route points to the destination SIP server to which the SIP call is routed. Note that the **phone-exact** entry applies a specific session configuration from the **session-config-pool**.

CLI Session

```
NNOS-E> config vsp
config vsp> config dial-plan
config dial-plan> config route E911
Creating 'route E911'
config route E911> set request-uri-match phone-exact 19788235233
config route E911> set action forward
config route E911> set peer server "vsp\enterprise\servers\sip-gateway
pstn"
config route E911> set session-config-pool-entry vsp
session-config-pool entry e911
config route E911> return
```

```
config dial-plan> config route acme packet
Creating 'route acme packet'
config route acme packet> set request-uri-match phone-prefix 19788235
config route acme packet> set action forward
config route acme packet> set peer server "vsp\enterprise\servers
    sip-gateway\acme packet"
config entry acme packet> return

config dial-plan> config route CompanyA
Creating 'route CompanyA'
config route CompanyA> set request-uri-match domain-suffix
    companyA.com
config route CompanyA> set action forward
config entry CompanyA> set peer server "vsp\enterprise\servers
    sip-gateway\gateway1"
```

For a complete description of the **request-uri-match** types (**phone-exact**, **phone-prefix**, **domain-suffix**, etc.) and their precedence when matching SIP requests, refer to the *Net-Net OS-E – Objects and Properties Reference*.

Inbound and Outbound Call Normalization

Call normalization is the process of changing all or a portion of a SIP URL so that the SIP call is routed to a particular destination. Normalization applies to both **inbound** and **outbound** SIP phone calls using the **host-normalization** properties. Inbound call normalization object properties apply to calls directed to destinations behind the OS-E system; outbound call normalization object properties apply to calls that pass through the OS-E destined for a destination outside of the network.

The **host-normalizations** property specifies when and how to change the host portion of an INVITE URL to match the domain name of the server configured for this dial-plan entry. The OS-E uses this property for users not listed in its registration table. (If the user is registered, the OS-E does a look up in the registration table to determine proper normalization.) When normalization occurs, the OS-E changes the specified portion of the header or URI to match the server's domain name.

The following CLI session applies normalization to inbound calls from a provider by changing

- The USER field of the request URI with the **request-user** property
- The USER field of the TO URI with the **to-user** property
- The USER field of the FROM header with the **from-user** property

CLI Session

```
NNOS-E> config vsp
config vsp> config dial-plan
config dial-plan> config route inbound
Creating 'route inbound'
config route inbound> set host-normalizations request-uri
config route inbound> set request-user replace-with string
config route inbound> set to-user prepend string
config route inbound> set from-user strip-off string
```

Operating On the SIP FROM Header

If a match occurs on the destination field of the URI, you can perform a second-level lookup on the FROM header using the **source** object. The OS-E selects the destination server from the configuration and applies a specific session configuration from the **session-config-pool**, if specified, or from the **session-config** under the **source** object.

Inbound and outbound call normalization, as described in the previous section, can be configured for FROM header matching strings.

CLI Session

```
NNOS-E> config vsp
config vsp> config dial-plan
config dial-plan> config route for978
Creating 'route for978'
config route for978> config source sip:1978.*
Creating 'source sip:1978.*'
config source sip:1978.*> set action forward
config source sip:1978.*> set source from-uri
config source sip:1978.*> set session-config-pool-entry vsp
    session-config-pool entry 1978.*
config source sip:1978.*> set peer server "vsp\enterprise\servers
    sip-gateway\pstn

config source sip:1978.*> return
config route for978> config inbound
config inbound> set host-normalizations from-header 1978.*
config inbound> set to-user no
```

Configuring the Dial-Plan Source Route

The **source-route** object operates the same way as the **route** object, as described above, except that the call routing/forwarding decision is based on the source IP address in the IP packet header rather than the request URI in the SIP message. Anytime a SIP call originates from this IP source route, a routing decision is made (based on the configuration) that forwards the call to a SIP destination server. This means that SIP calls from multiple sources, based on the source IP address in the IP packet header, can then be routed to different SIP destination servers (sip-gateway, pstn-gateway, etc.).

The **source-match** property operates on the specified IP address for the following **source-match** property options:

- host
- ipnet
- server
- carrier
- gateway

The trunk option operates on the SIP message Contact header rather than the source IP address.

Like the **route** object, inbound and outbound call normalization is supported on SIP traffic matching the **source-route** configuration.

CLI Session

```
NNOS-E> config vsp
config vsp> config dial-plan
config dial-plan> config source-route fromIP
Creating 'source-route fromIP'
config source-route fromIP> set source-match host 123.45.1.34
config source-route fromIP> set action forward
config source-route fromIP> set peer server
    "vsp\enterprise\servers\sip-gateway pstn"
config source-route E911> set session-config-pool-entry vsp
    session-config-pool entry fromIP
config source-route fromIP> return
```

Configuring Dial-Plan Routing Arbitration

Routing arbitration allows you to configure different cost-based routing algorithms to be used in selecting where the OS-E forwards inbound SIP calls. For a given destination SIP server, multiple carriers may be available to route the call. When the OS-E receives a SIP call, it makes a determination where to forward the call (to the next hop) base on a routing arbitration decision.

The OS-E dial-plan routing arbitration uses the **vsp\carriers\carrier** and the **vsp\carriers\hunt-group** (group of carriers) configuration to determine the carrier, gateway, or trunk to connect the call to the destination SIP server.

The routing algorithms that are available for routing arbitration are:

- Customer-preferred carrier—Using the **most-preferred** arbitration rule.
- Bandwidth utilization—Using the “most available” link; configured with **least-load** or **least-calls** rule.
- IP QoS—Using the carrier link having the best quality-of-service metrics; configured with the **trunk-qos** rule.
- Cost—Using the carrier with the lowest routing cost metric to that destination; configured with the **least-cost** rule.

The following CLI session configures best-match, route-server routing for inbound traffic matching the *carrier123.net* domain. The OS-E will consider two routes in parallel from the carrier options included in **voip\hunt-group\group1**. The **session-config-pool** entry named *lc-route-arbitration* controls the call session enroute to the destination.

CLI Session

```
config> config vsp dial-plan
config dial-plan> config arbiter leastCost1
Creating 'arbiter leastCost1'
config arbiter leastCost1> set arbiter-apply best-match
config arbiter leastCost1> set max-call-hunting-options 2
config arbiter leastCost1> set call-hunting-type parallel
config arbiter leastCost1> set rule least-cost
config arbiter leastCost1> set session-config-pool-entry vsp
    session-config-pool entry lc-route-arbitration
config arbiter leastCost1> set subscriber-match domain-exact
    carrier123.net
config arbiter leastCost1> return
```

```
config dial-plan>
config dial-plan> config route carrier123
Creating 'route carrier123'
...carrier123> set peer hunt-group "vsp\voip\hunt-group\group1 "
```

Configuring a Dial Route For the Request URI

Based on the arbitration rules that you configure, you may want to direct the dial route for certain SIP request URIs (FROM) to a specific destination. This allows you to direct subscriber request URIs to a specified destination using the arbitration metrics (QoS, least-cost, most-preferred, etc.).

The settings are the same as those for **arbiter** object with one difference; you set the **request-uri-expression** property on which to match when the OS-E performs route arbitration.

The following CLI session performs route arbitration using a **most-preferred** rule setting for all calls matching *sip:1413*@.** to a 617 destination, using the **session-config-pool** configuration.

CLI Session

```
config> config vsp dial-plan
config dial-plan> config arbiter calls-to-617
Creating 'arbiter calls-to-617'
config arbiter calls-to-617> config destination 617
config destination 617> set arbiter-apply best-match
config destination 617> set max-call-hunting-options 2
config destination 617> set call-hunting-type none
config destination 617> set rule most-preferred
config destination 617> set session-config-pool-entry vsp
    session-config-pool entry 617
config arbiter nextNet> set request-uri-expression sip:1413*@.*
config arbiter nextNet> return
config dial-plan>
```

Configuring Dial-Plan Normalization

Use the **normalization** object to simplify and speed up inbound and outbound call-lookups in the call routing tables. As described earlier, call normalization is the process of changing all or a portion of a SIP URL so that the SIP call is routed to a particular destination. Normalization applies to both **inbound** and **outbound** SIP phone calls using the **normalization** properties. Inbound call normalization object properties apply to calls directed to destinations behind the OS-E; outbound call normalization object properties apply to calls that pass through the OS-E to a destination outside of the network.

The following CLI session normalizes a SIP phone call to *sip:5008@acmepacket.com* by changing the lookup address to *sip:19788235008@acmepacket.com*.

CLI Session

```
config> config vsp dial-plan
config dial-plan> config normalization 978
config normalization 978> set match phone-exact
    sip:5008@acmepacket.com
config normalization 978> set alter-phone-scheme no
config normalization 978> config inbound
config inbound> set host-normalizations request-uri
config inbound> set request-user replace-with
    sip:19788235008@acmepacket.com
```

Configuring the Dial-Prefix

The **dial-prefix** object allows you to apply a custom session configuration based on a dial prefix found in either the To or REQUEST URI of the SIP header. If you set a prefix, such as *61 for an originating SIP phone call, your session configuration can initiate a session action, such as call recording, when the OS-E detects that dial prefix in the SIP header.

If the OS-E detects the specified dial prefix, it strips off the prefix and applies a session-config object to the session. This is either the session configuration within the dial-prefix object, or a **session-config** entry from the **session-config-pool** that you reference with **session-config-pool-entry** property.

The following CLI session strips off the *61 dial prefix and runs the unique *61 session configuration.

CLI Session

```

NNOS-E> config vsp
config vsp> config dial-prefix *61
config dial-prefix *61> set dial-prefix *61
config dial-prefix *61> set session-config-pool-entry vsp
      session-config-pool entry *61
  
```

Viewing the Call Routing Tables

To display information about how the OS-E routes SIP call INVITEs, use the **show dial-plan** and the **show call-routing** commands to display the entries in the dial-plan and in the call routing table. The dial-plan table contains the configured entries from the vsp/dial-plan object, while the call-routing table contains the call forwarding lookup entries with an active SIP server. If that server becomes unavailable, the entry is removed from the call routing table.

You execute the **show dial-plan** and the **show call-routing** commands from the top-level OS-E prompt, or from the OS-E Management System **Status** tab. Each table displays the following information:

- plan-name
- type
- destination URL expression
- subscriber-match
- peer name
- fwd

CLI Session

```

NNOS-E> show call-routing
  
```

plan-name	type	destination-url-expression	sub-match	peer-name	fwd
Mass978	phone	sip:1978.*@.*	.*	HiTechGW	0
Acmepacket	phone	sip:19788235.*@.*	.*	Acmepacket	0
JohnSmith	phone	sip:19788235200.*@.*	.*	Mgt	0

In this example,

1. A call to *John Smith* at phone number *19788235200* triggers a call lookup that matches the dial plan entry named *JohnSmith*.
2. A call to phone number *19788235218* triggers a call lookup that matches the dial plan entry named .
3. A call to phone number *19784191234* triggers a call lookup that matches the dial plan entry named *Mass978*.
4. If the *Mgt* server becomes unavailable, the matching route for *John Smith* is removed from the call routing table and future calls to *John Smith* will use the SIP server named .

Dial-plan Related Show Commands

The following show commands can provide useful information about the dial-plan and routing arbitration configuration:

- **show call-normalization**
- **show carrier-rate-plans**
- **show call-routing**
- **show dial-plan**
- **show hunt-groups**
- **show hunt-group-options**
- **show routing-arbitration**
- **show rules**

For detailed information on the **show** commands, refer to the *Net-Net OS-E – Objects and Properties Reference*.

Chapter 4. Configuring Location Services

About This Chapter

The chapter describes the location service on the OS-E system. Configuring the location service allows the OS-E to proxy SIP REGISTER requests. Location and contact information, such as addresses of record and their associated tags, are exchanged between the SIP registrars.

How the Net-Net OS-E Stores Location Information

Location and registration information is stored in a location service database and is shared with registration peers. This allows SIP callers to reach (and locate) SIP call recipients who have one or more addresses-of-record stored in the database. The call recipients also use location and registration services to call back the originating SIP caller using SIP INVITE messages.

A SIP call is preceded by a REGISTER request. These requests add and remove bindings between addresses of record (AORs) and contact addresses. The OS-E learns location records in different ways, and stores the appropriate source information with the record, as listed below.

If the record is...	It's source is...
learned from a client	Client
statically configured	Static
reloaded from the location database	Database
result of proxy registration	Proxy

If the record is...	It's source is...
learned from a registrar peer	Peer
result of a DNS lookup	DNS

Address-Of-Record Static Bindings

The static address-of-record bindings associate SIP recipients to specific domain names, as well as provide contact information for the SIP recipients. Static bindings are those that you manually configure using the **address-of-record** configuration object as opposed to address-of-record bindings received from a registration server, learned from a SIP client, or defined by configured policies.

Address-of-record bindings map an incoming SIP or SIPS URI, such as sip:bob@company.com to one or more URIs that are more direct to that user, such as the extended sip:bob@marketing.company.com entry.

When specifying the SIP or SIPS (SIP secure) address-of-record, enter the word sip or sips followed by a colon (:), followed by the uniform resource identifier (URI) string associated with the SIP user.

Tags Associated With an Address Of Record

When the OS-E receives a REGISTER request with no tag, it queries the vsp enterprise directory service for the AOR. The directory service returns all addresses of record for that user. Each record has a tag associated with it. The OS-E then sends a proxy registration for each record, based on the tag.

For example, if a REGISTER request comes in for user Tim, the enterprise directory might return the following:

Address of record	Tag
sip:2408882002@as.broadworks.net	bw
sip:tim@companyXYZ.com	avaya
sip:2002@companyXYZ.com	cov

The OS-E passes the registration `sip:2408882002@as.broadworks.net` to the BroadWorks server, the registration `sip:tim@companyXYZ.com` to the Avaya server in the `companyXYZ.com` domain, and the registration `sip:2002@companyXYZ.com` local, because `cov` is the configured tag.

CLI Session

```
config> config vsp enterprise
config enterprise> config directories
config directories> config active-directory companyXYZ
Creating 'active-directory companyXYZ'
config active-directory companyXYZ> set tag cov
config active-directory companyXYZ> config user-attributes
config user-attributes> set name ipphone
config user-attributes> set name extension
config user-attributes> return
config active-directory companyXYZ> set domain companyXYZ.com
config active-directory companyXYZ> set host 192.168.1.84
config active-directory companyXYZ> set username name
config active-directory companyXYZ> set password-tag password

config directories> config phantom bw
Creating 'phantom bw'
config phantom bw> set tag bw
config phantom bw> set domain as.broadworks.net
config phantom bw> set parent-directory vsp enterprise directories
    active-directory companyXYZ
Creating 'vsp\enterprise\directories\active-directory companyXYZ'
```

Configuring the Location-Service

This section provides a sample location service configuration using the CLI. The configuration, as illustrated in the image below.

- Configures static address-of-record bindings for the SIP users named *Rob* and *Alice*.
- Configures the location service database and sets the OS-E to save address-of-record bindings to the SQL database on disk (by setting the **persistent** property to *true*.)

CLI Session

```
NNOS-E> config vsp location-service
```

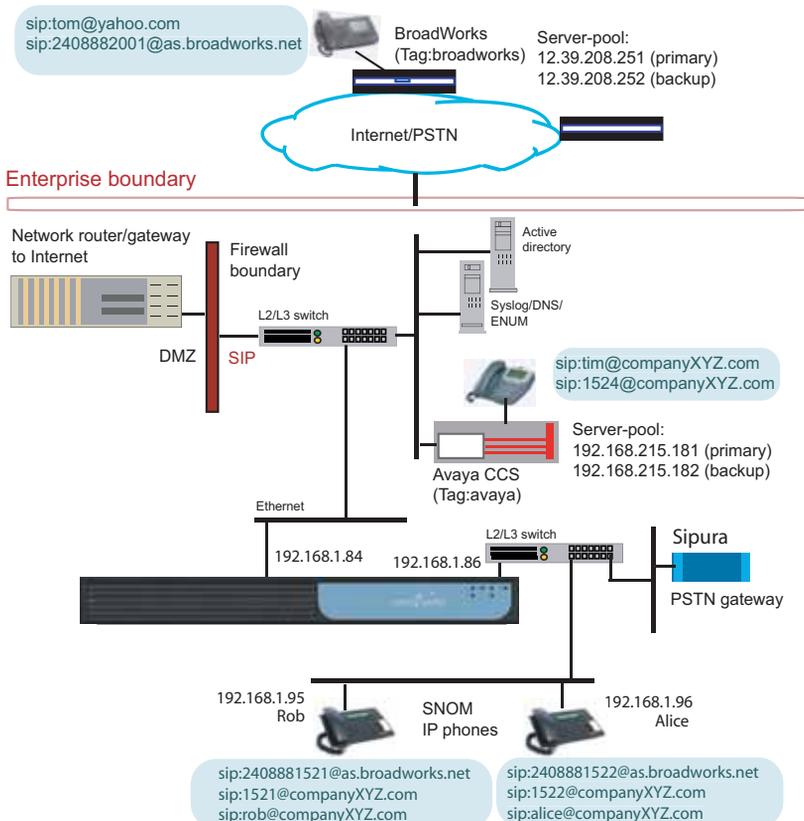
```

config location-service> config address-of-record
  sip:rob@companyXYZ.com
Creating 'address-of-record sip:rob@companyXYZ.com'
config address-of-record sip:rob@companyXYZ.com> set contact
  sip:192.168.1.95 UDP 5060
config address-of-record sip:rob@companyXYZ.com> return

config location-service> config address-of-record
  sip:alice@companyXYZ.com
Creating 'address-of-record sip:alice@companyXYZ.com'
config address-of-record sip:rob@companyXYZ.com> set contact
  sip:192.168.1.96 UDP 5060
config address-of-record sip:rob@companyXYZ.com>> return

config location-service> config database
config database> set persistent true

```



Chapter 5. Configuring SIP Registration Services

About This Chapter

This chapter describes SIP registration, the mechanism that instructs the OS-E routing action (accept, delegate, forward, redirect, tunnel, discard, block) on received SIP REGISTER requests.

About SIP Registration and Registration Plans

The OS-E system can forward, delegate, proxy, redirect, or block client registrations. In networks where proxy registration may be desired, the OS-E accepts SIP REGISTERs on behalf of an upstream destination SIP registrar. In carrier and service-provider networks, SIP REGISTERs can be delegated directly to the designated SIP registrar.

When the OS-E receives a REGISTER request from a SIP client, the OS-E looks in the registration-plan table for a configuration match with the SIP client. If a match exists in the registration plan table, then the REGISTER is forwarded, delegated, proxied or redirected to the destination server as configured in the matching registration plan.

If a match does not exist in the registration plan, then the REGISTER is checked against the local registration service. If the local registration service determines the REGISTER is accepted, the OS-E responds with a “200 OK” message to the client. Otherwise, the OS-E responds with a “600 Decline” message to the client. By default, SIP registration on the OS-E is disabled, blocking SIP REGISTER sessions from being established with the OS-E or to a delegated SIP registrar.

If the destination server is busy or unavailable, the OS-E acts as the SIP proxy and can forward the call using a configured PSTN gateway. Before configuring SIP registration, you should configure the appropriate dial-plans and configure the OS-E location service. Refer to the following chapters:

- Chapter 3, “Configuring Dial Plans”
- Chapter 4, “Configuring Location Services”

The registration plan is a prefix/suffix-match based table to specify how to process a registration. Generally, the most specific match is always preferred. However, you can create a configuration so that a particular phone number or URI can use a less specific match. The OS-E provides a number of match patterns, as listed below in the order of the most specific to the least specific match, with an example of each.

- URI (sip:john@compXYZ.com)
- Directory (IP phone directory, active directory)
- Phone prefix (978; any phone number with prefix 978)
- Phone exact
- Domain suffix (compXYZ.com, broadworks.net)
- Domain exact
- Source IP prefix (61.21.32.0/24)
- Default (or anything else)

The OS-E provides the following SIP REGISTER actions:

- **Accept**—Accepts a client registration as a SIP registrar
- **Delegate**—Delegates registration to another registrar,
- **Forward**—Forwards registration to another registrar
- **Redirect**—Instructs the SIP client to send subsequent registrations to another registrar
- **Block**—Rejects registrations
- **Use-policy**—Registration is subject to a policy lookup

You can configure domain-aware-phones so that a certain range of phone numbers are subject to a domain-suffix match rather than a phone-prefix match, as configured in the following CLI session using the registration-plan **settings** object:

CLI Session

```
NNOS-E> config vsp registration-plan settings
config settings> set domain-aware-phone-expression \d\d\d
```

To display all registration plans, enter the following show command:

```
NNOS-E> show registration-plan
```

A registration plan becomes active when the referenced peer is up. To display all active registration plans, enter the following command:

```
NNOS-E> show registration-routing
```

Enabling the Local Registration Service

The following CLI session enables the OS-E local registration service to accept SIP REGISTER traffic.

CLI Session

```
NNOS-E> config vsp
config vsp> set local-identity compXYZ.com
config registration-service> set admin enabled
```

This configuration adds the "compXYZ.com" entry to the registration plan, matching the domain name with an action of Accept. It enables the OS-E to accept any registration for URI "sip:.*@compXYZ.com".

To display whether the registration-service is configured or enabled, enter the following command:

```
NNOS-E> show registration-service
```

If needed, you can configure other static entries to the registration plan. For example:

```
NNOS-E> config vsp registration-plan
config registration-plan> config route 978
config route 978> set to-uri-match phone-prefix 978
config route 978> set action accept
config route 978> config session-config
config session-config> config authentication
config authentication> set mode RADIUS compXYZ-group
```

This configuration enables the OS-E to accept any registration for URI *sip:978.*@.**. Additionally, for any phone prefix 978, the OS-E challenges clients in order to authenticate and authorize client registrations.

To display local registration statistics, enter the following command:

```
NNOS-E> show registration-status
```

Configuring Registration-Plan Settings

The registration-plan **settings** object allows you to configure domain-aware phones so that a specific range of numbers are not subject to phone-prefix match. Enter a regular expression for the **domain-aware-phone-expression** property to create phone numbers that are matched on the suffix found in the URI instead of the phone number. For example, if the URI contained a three-digit extension, and the settings expression is set to `\d\d\d`, the OS-E matches on domain suffix instead of phone number, as shown in the CLI session below.

CLI Session

```
NNOS-E> config vsp
config vsp> config registration-plan
config registration-plan> config settings
config settings> set domain-aware-phone-expression \d\d\d
```

Delegating Registrations

When delegating SIP REGISTERs to another registrar, the OS-E

1. Receives the REGISTER from a type of transport
2. Normalizes the REGISTER, manipulating some SIP headers such as the Contact header
3. Sends the REGISTER to another registrar through the same or different type of transport

The OS-E delegates the user authentication and authorization to the upstream registrar. Registration delegation allows existing SIP infrastructures to participate with the OS-E system for secure communication and improved session control.

CLI Session For the Upstream Registrar Configuration

```
NNOS-E> config vsp enterprise servers sip-gateway broadworks
Creating '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> config server-pool
config server-pool> config server primary
Creating 'server primary'
config server primary> set host 12.39.208.251
config server primary> return
config server-pool> config server backup
Creating 'server backup'
config server backup> set host 12.39.208.252
config server backup> return
config server pool> return
```

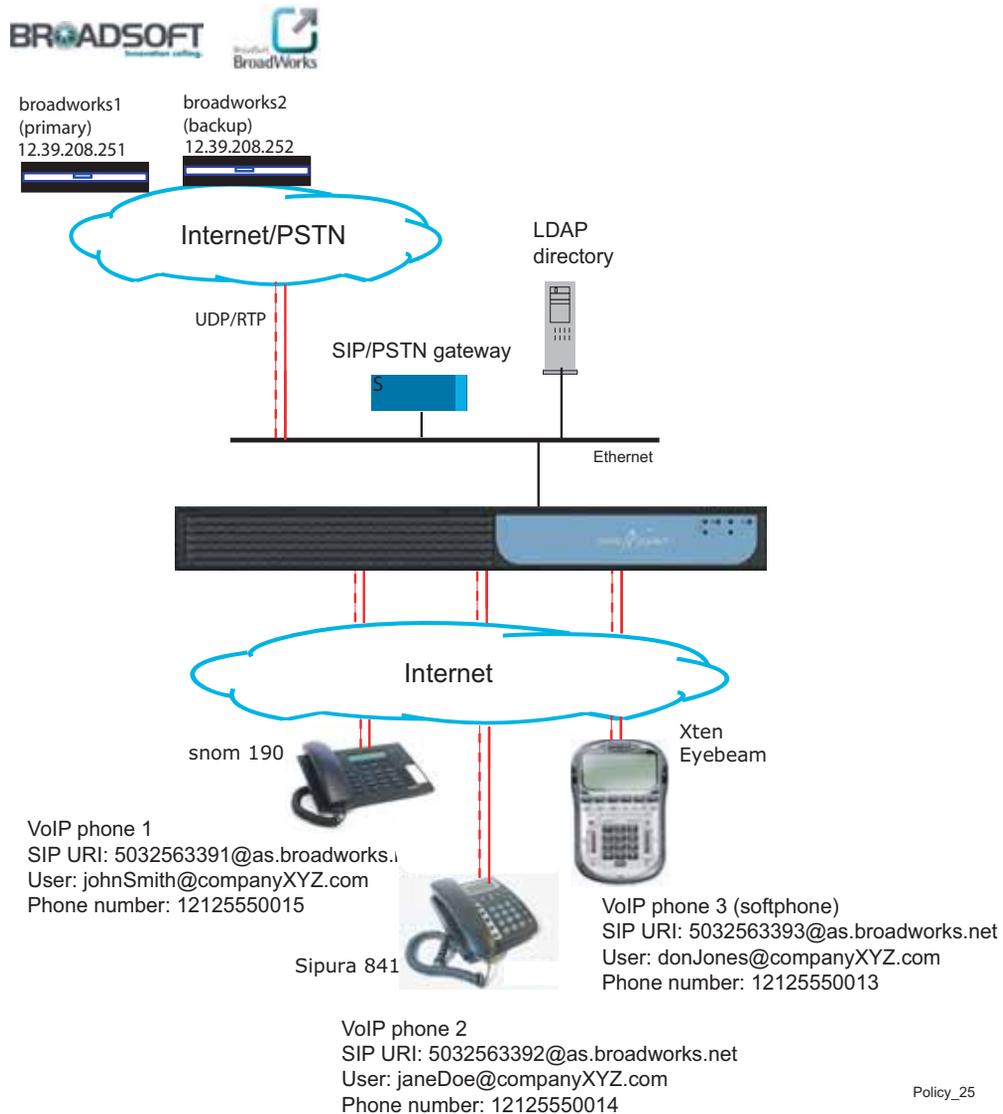
CLI Session For the OS-E Registration-Plan Configuration

```
NNOS-E> config vsp registration-plan
config registration-plan> config route broadworks
Creating 'route broadworks'
config route broadworks> set to-uri-match domain-suffix broadworks.net
config route broadworks> set action delegate
config route broadworks> set alter-contact local-host
config route broadworks> set peer server "vsp enterprise servers
    sip-gateway broadworks"
config route broadworks> return
```

There are two ways to alter the contact header: using **local-host** or **local-maddr**. If the URI in a REGISTER is *sip:9788235233@compXYZ.com* and the OS-E interface address is *192.168.100.1*, then the OS-E alters the contact header as follows:

- **local-host**—*sip:9788235233@192.168.100.1:5060;transport=UDP*
- **local-maddr**—*sip:9788235233@compXYZ.com:5060;transport=UDP;maddr=192.168.100.1*

The sample registration delegation configuration for the Broadworks application is illustrated in the following image.



In the example network configuration above, the OS-E delegates SIP registrations to the BroadWorks server by

- Configuring the names of the SIP registrar peers; *primary* and *backup*.

- Setting the domain properties associated with each registrar peer, including the domain-name and domain-alias.
- Configuring a server pool associated with the registrars, where the pool has a primary and a backup SIP registrar should one of the registrar peers become unavailable. Each server is identified by host IP (and transport, maximum number of concurrent calls, and the maximum number of calls allowed to be setup for each peer session).

Refer to the *Net-Net OS-E – Objects and Properties Reference* for information on other properties that you can set with SIP registrars and server pools.

To display registrar peer status, enter the following command:

```
NNOS-E> show sip-peers -v
```

To display registration delegation statistics, enter the following command:

```
NNOS-E> show registration-delegation
```

Authenticating With the Upstream Registrars

For the OS-E to delegate SIP registrations and addresses-of-record to an upstream registrar peer, (such as Broadworks) the peer must be configured with matching user and password-tag credentials. This means that the registrar peer authenticating REGISTER sessions must have a matching user name and password-tag associated with the other peer in its configuration.

CLI Session

```
NNOS-E> config vsp enterprise servers sip-gateway broadworks
...broadworks> set domain as.broadworks.net
...broadworks> set domain-alias broadworks.net
...broadworks> set domain-subnet 12.39.208.251/24
...broadworks> set domain-subnet 12.39.208.252
...broadworks> config server-pool
...server-pool> config server primary
Creating 'server primary'
config server primary> set host 12.39.208.251
config server primary> set transport tls "vsp tls certificate name"
config server primary> return
...server-pool> config server backup
Creating 'server backup'
config server backup> set host 12.39.208.252
config server backup> set transport tls "vsp tls certificate name"
config server backup> return
```

```
...server-pool> return
...broadworks> set user cxc
...broadworks> set password-tag broadworks
```

Pinging Upstream Registrars For Availability

The **failover-detection** and **ping-interval** properties configure the OS-E to monitor the availability of the registrar peers. By setting the **failover-detection** property to auto with a configured time interval, the OS-E polls peer registrars to see if they are available to accept registration and address-of record updates.

The OS-E automatically places all SIP registrar peers defined in the server-pool configuration into a monitoring pool. Because each SIP server has an associated “order,” the server with the least order is the preferred *primary*. The other SIP registrar peers are the *backup* servers.

The OS-E monitors registrar peers using the configured ping interval setting (in seconds). When all peers have responded to the OS-E pings, the peer designated as the primary assumes the responsibility for accepting registrations and AORs from the OS-E.

If the primary is in the UP stated as indicated by the monitoring process, then registrations and INVITEs are sent to the primary. If primary is down while the backup is up, then registrations and INVITEs are forwarded to the backup. When the primary becomes available again later, registrations and INVITE messages redirect back to the primary.

If both the primary and backup registrar peers are unavailable, SIP INVITE messages are switched locally if the To: header is addressed to a SIP phone connected to the OS-E directly (where it has a location binding with the OS-E). If the To: header is addressed to another location, the INVITE is forwarded to a PSTN gateway:

CLI Session

```
NNOS-E> config vsp enterprise servers sip-gateway broadworks
...broadworks> set domain as.broadworks.net
...broadworks> set domain-alias broadworks.net
...broadworks> set domain-subnet 12.39.208.251
...broadworks> set domain-subnet 12.39.208.252
...broadworks> set match tight
...broadworks> set peer-identity sip:broadworks@compXYZ.com
...broadworks> set failover-detection auto
...broadworks> set ping-interval 20
```

To interactively check if a SIP server or host is reachable through SIP, use the **sip ping** command, as shown in the session below:

```
NNOS-E> sip ping as.broadworks.net UDP 5060
```

To include SIP servers into the monitoring pool, use the **sip server-monitor** command:

```
NNOS-E> sip server-monitor 12.39.208.251 UDP 5060  
NNOS-E> sip server-monitor 12.39.208.252 UDP 5060
```

To display the availability status of the SIP servers that you are monitoring, execute the **show sip-server-availability** command.

```
NNOS-E> show sip-server-availability
```

To remove SIP servers from the monitoring pool, execute the **sip server-unload** command.

```
NNOS-E> sip server-unload 12.39.208.251 UDP 5060  
NNOS-E> sip server-unload 12.39.208.252 UDP 5060
```

To disable SIP pings to a registrar peer, set the **failover-detection** property to none.

```
NNOS-E> config vsp enterprise servers sip-gateway broadworks  
config sip-gateway broadworks> set failover-detection none
```

Creating Registration-Plan Routes

Each registration-plan describes how to treat a matched SIP REGISTER session. Registration-plan entries are stored in a registration routing table that the OS-E uses to look up and match REGISTER sessions from peer registrars.

The OS-E uses a longest-prefix match lookup to match the most specific entry. If a peer becomes unavailable, the OS-E finds the next longest match and forwards the call to that peer. For detailed information on how the OS-E performs longest-prefix matching, refer to the *Net-Net OS-E – Objects and Properties Reference*.

The following CLI session creates a registration-plan that delegates SIP REGISTER requests to the *broadworks.net* domain.

CLI Session

```
NNOS-E> config vsp registration-plan  
config registration-plan> config route broadworks  
Creating 'route broadworks'
```

```
config route broadworks> set to-uri-match domain-suffix broadworks.net
config route broadworks> set action delegate
config route broadworks> set peer server "vsp enterprise servers
    sip-gateway broadworks"
```

Proxying Registrations

In proxy registration, the OS-E accepts SIP REGISTERs locally and saves bindings to the location-service database. The OS-E also originates SIP REGISTERs on behalf of clients and sends the SIP REGISTERs to the upstream registrar.

For a single REGISTER, the OS-E can proxy the registration to many registrar peers as long as each registrar peer has a matching registration plan.

The OS-E periodically (usually once a day) downloads all address-of-record (AOR) bindings to the registrar peers that have a matching registration-plan. If the primary registrar becomes unavailable, the OS-E downloads all AOR bindings to the backup registrar, with an AOR matching the registration plan to the unavailable registrar.

Sample Proxy Registration Configuration

The following CLI session configures the OS-E proxy registration, the upstream registrar and the registration-plan.

CLI Session

You need to enable **registration-proxy** under the VSP object for performance considerations.

```
NNOS-E> config vsp
config vsp> set registration-proxy enabled
config vsp> return
```

Configure the upstream registrar peers.

```
NNOS-E> config vsp enterprise servers 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 peer-identity sip:as.broadworks.net
config sip-gateway broadworks> config registration-proxy
config registration-proxy> return

config sip-gateway broadworks> set user cxc
```

```

config sip-gateway broadworks> set password-tag broadworks
config sip-gateway broadworks> set peer-max-interval 38400
config sip-gateway broadworks> set peer-min-interval 60
config sip-gateway broadworks> config server-pool
config server pool> config server primary
config server primary> set host 12.39.208.251
config server primary> return
config server pool> config server backup
config server backup> set host 12.39.208.252
config server backup> return
config server-pool> return
config sip-gateway broadworks> top

```

Configure the registration-plan.

```

config> config vsp registration-plan
config registration-plan> config route broadworks
Creating 'route broadworks'
config route broadworks> set to-uri-match domain-suffix broadworks.net
config route broadworks> set action accept
config route broadworks> set alter-contact local-host
config route broadworks> set peer server "vsp enterprise servers
    sip-gateway broadworks"
config route broadworks> return

```

Configure the registration-plan proxy.

```

config registration-plan> config proxy compXYZ.com
config proxy compXYZ.com> set admin enabled
config proxy compXYZ.com> set uri-match domain-exact broadworks.net
config proxy compXYZ.com> set priority 100
config proxy compXYZ.com> config peer 1
config peer 1> set peer server "vsp enterprise servers sip-gateway
    broadworks"

```

Client Authentication Over Proxy Registration

To the SIP client, the OS-E may challenge the client for authentication and then authorize client registration after a valid response from the client. From the upstream registrar, the OS-E may receive a challenge. In this case, the OS-E originates the response based on common credentials with the upstream registrar.

For the OS-E to respond to an authentication challenge from upstream registrar, you need to configure a single credential for the registrar peer:

```

NNOS-E> config vsp enterprise servers sip-gateway broadworks
config sip-gateway> set user cxc
config sip-gateway> set password-tag broadworks
config sip-gateway> top

```

Upon receipt of a “401 Unauthorized” message after sending a REGISTER to an upstream registrar (sip:as.broadworks.net), the OS-E performs the following:

1. Retrieves credentials from the peer configuration.
2. Requests a response from the authentication process.
3. Constructs an authorization header.
4. Sends a new REGISTER to the upstream peer.

Keep in mind that you must have matching credentials (such as user: cxc, password-tag: broadworks) configured on the BroadWorks application server.

To display registration proxy statistics, enter the following command:

```
NNOS-E> show registration-proxy
```

To debug, type

```
NNOS-E> registration client-save (to view the registration-clients)
Success!
NNOS-E> show registration-clients
NNOS-E> registration client-clear (to clear registration-clients)
Success!
```

To query upstream registrar for an AOR binding, enter the following command:

```
NNOS-E> registration query sip:2408881521@as.broadworks.net
sip:as.broadworks.net
```

The expiration time should decrease every second in the query.

Address-Of-Record Bindings

AOR bindings are processed in proxy mode and in back-to-back (B2B) mode. In proxy mode, the OS-E forwards location bindings to its upstream registrar AS IS, with no changes to the Contact header. This means client locations are visible to the peer and the peer can communicate with the client directly.

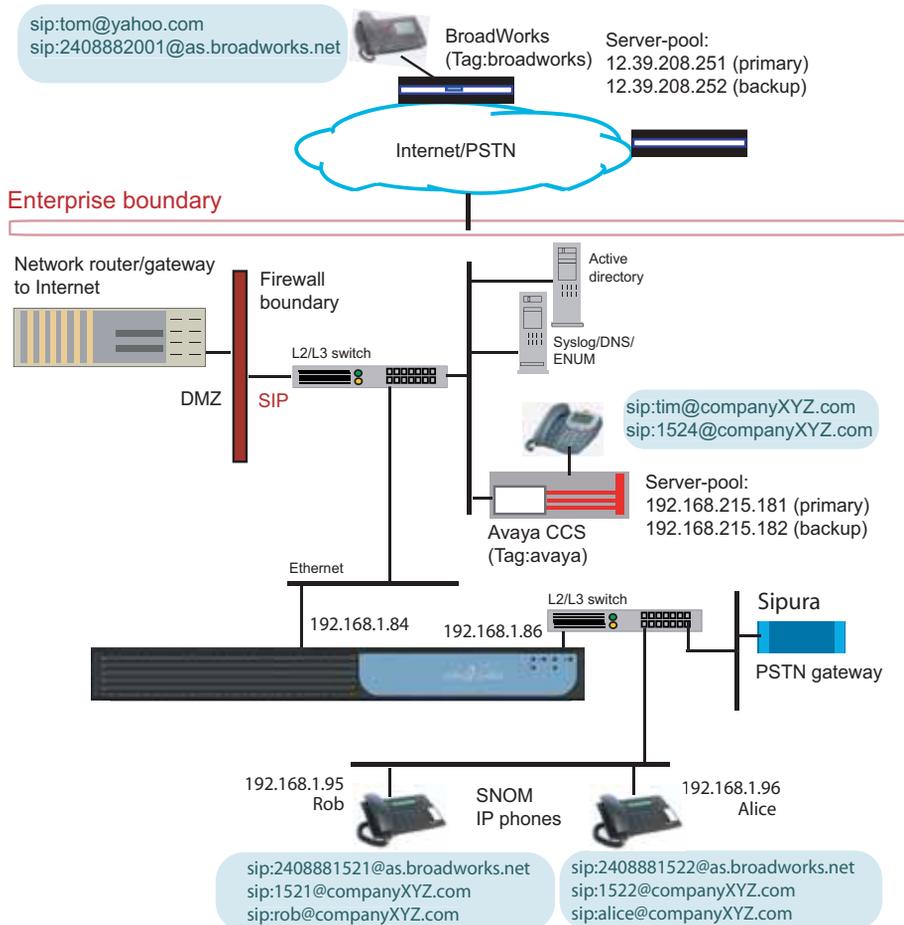
In the B2B mode, the registration clients are hidden from the upstream registrar. The OS-E alters the Contact as configured in registration-plan.

There are two ways to alter the Contact header: **local-host** or **local-maddr**. If the URI in a REGISTER is *sip:9788235233@compXYZ.com* and the OS-E interface address is *192.168.100.1*, then the Contact header is altered as follows:

- **local-host**—sip:9788235233@192.168.100.1:5060;transport=UDP
- **local-maddr**—
 sip:9788235233@compXYZ.com:5060;transport=UDP;maddr=192.168.100.1

Proxy Registration Network Example

The following image illustrates a sample network using proxy registration, followed by some sample scenarios.



Action: Proxy Registration

1. Rob's SNOM phone registers with the OS-E. The URI is sip:rob@compXYZ.com, and contact is 192.168.1.95 (UDP, 5060).
2. The OS-E queries the Active Directory, and retrieves all Rob's AORs:
 - sip:rob@compXYZ.com (tag:xyz)
 - sip:2408881521@as.broadworks.net (tag: bw)
 - sip:1521@compXYZ.com (tag: avaya).
3. In response to Rob's "bw" tag, the OS-E sends a proxy registration to the BroadWorks server, with sip:2408881521@as.broadworks.net as the AOR, and UID.bw-uid@192.168.1.86 (the OS-E system's address) as the contact.
4. In response to Rob's "avaya" tag, the OS-E sends a proxy registration to the Avaya PBX, with sip:1521@compXYZ.com as the AOR, and UID.avaya-uid@192.168.1.86 as the contact.
5. Similarly, as a result of Alice's registration with the OS-E, it sends proxy registrations to:
 - BroadWorks server, with sip:2408881522@as.broadworks.net as the AOR and UID.bw-uid@192.168.1.86 (OS-E's address) as the contact.
 - Avaya PBX, with sip:1522@compXYZ.com as the AOR and UID.avaya-uid@192.168.1.86 as the contact.

Action: Rob Calls Alice at BroadWorks Phone Number 2408881522

1. The OS-E determines that the destination tag is bw (BroadWorks).
2. The OS-E changes Rob's From header to sip:2408881521@as.broadworks.net and forwards the call to the BroadWorks server.
3. BroadWorks forwards the call back to the OS-E and it forwards the call to Alice.

Action: Rob Calls Tom at BroadWorks Phone Number 2408882001

1. The OS-E determines that the destination tag is bw (BroadWorks).
2. The OS-E changes Rob's From header to sip:2408881521@as.broadworks.net and forwards the call to the BroadWorks server.
3. BroadWorks forwards the call to Tom.

Action: Rob Calls Tim at compXYZ x1524

1. The OS-E determines that the destination tag is avaya.
2. The OS-E changes Rob's From header to sip:1521@compXYZ.com and forwards the call to Avaya's PBX.
3. Avaya's PBX forwards the call to Tim.

Action: Rob Calls Alice at sip:alice@compXYZ.com

1. The OS-E determines that the destination tag is xyz (compXYZ).
2. The OS-E does not change Rob's From header and forwards the call to Alice directly.

Performing Other Actions On SIP Registrations

In addition to delegating and proxying (accepting) SIP REGISTER sessions at the OS-E system, you can also create registration plans that forward, redirect, tunnel, discard, or block SIP REGISTER sessions.

Forwarding Registrations

If a SIP REGISTER request matches a registration plan with the action property set to **forward**, the OS-E makes no modifications to the session. The SIP session is forwarded to the target SIP registrar with no changes to the contact information.

If the **session-config** has the **registration/cache-lcs** property set to enabled, the OS-E caches the binding upon successful registration. The bindings cached from forwarded registrations can be used for proxy registration and subsequent call forking.

CLI Session

Configure the upstream server.

```
NNOS-E> config vsp enterprise servers lcs company
config lcs company> set domain company.com
config lcs company> config server-pool
config server-primary> configure server primary
Creating 'server primary'
config server primary> set host 192.168.215.54
```

```
config server primary> return
```

Configure the registration plan.

```
NNOS-E> config vsp registration-plan
config registration-plan> config route company
config route company> set to-uri-match domain-suffix company.com
config route company> set action forward
config route company> set alter-contact no
config route company> set peer server "vsp enterprise servers lcs
    company"
config route company> return
```

Redirecting Registrations

If a SIP REGISTER request matches a registration plan with the **action** property set to **redirect**, the OS-E redirects a client registration to an alternative registrar. In this case, the OS-E does the following:

1. Responds with “301 Redirect” message
2. Sets the Contact header to the alternate registrar's address, port and transport
3. Instructs the client to send subsequent registrations to the alternate registrar

CLI Session

Configure the upstream registrar.

```
NNOS-E> config vsp enterprise servers sip-gateway broadworks-backup
Creating 'sip-gateway broadworks-backup'
config sip-gateway broadworks-backup> set domain as.broadworks.net
config sip-gateway broadworks-backup> set domain-alias broadworks.net
config sip-gateway broadworks-backup> config server-pool
config server-pool> config server backup
Creating 'server backup'
config server backup> set host 12.39.208.252
config server backup> return

config server pool> config server primary
config server primary> set host 12.39.208.251
config server primary> return
config server-pool> return
config sip-gateway broadworks-backup>
```

Configure the registration-plan.

```
NNOS-E> config vsp registration-plan
config registration-plan> config route broadworks
```

```
config route broadworks> set to-uri-match domain-suffix broadworks.net
config route broadworks> set action redirect
config route broadworks> set alter-contact no
config route broadworks> set peer server "vsp enterprise servers
    sip-gateway broadworks-backup"
config route broadworks> return
config registration-plan>
```

Tunneling Registrations

With the registration-plan **action** property set to **tunnel**, the OS-E can create an OC client-to-LCS server tunnel over which SIP REGISTER traffic can be load balanced. An OC client-to-LCS server tunnel consists of two separate connections that the OS-E joins—a connection from the client to the OS-E and a connection from the OS-E to the server.

For additional information on OC client-to-LCS server tunnels, refer to the VSP settings object in the *Net-Net OS-E – Objects and Properties Reference*.

CLI Session

Configure the registration-plan for tunneling SIP REGISTER traffic.

```
NNOS-E> config vsp registration-plan
config registration-plan> config route oc-to-lcs
config route oc-to-lcs> set to-uri-match phone-prefix 617
config route oc-to-lcs> set action tunnel
config route oc-to-lcs> return
config registration-plan>
```

Discarding Registrations

With the registration-plan **action** property set to **discard**, the OS-E discards REGISTER requests matching the registration plan or AOR.

CLI Session

Configure the registration-plan for discarding SIP REGISTER traffic.

```
NNOS-E> config vsp registration-plan
config registration-plan> config route 978
config route 978> set to-uri-match phone-prefix 978
config route 978> set action discard
config route 978> return
config registration-plan>
```

Blocking Registrations

If a SIP REGISTER request matches a registration plan with the **action** property set to **block**, the REGISTER session is blocked unconditionally at the OS-E system with a “601 Declined” message. The OS-E takes no further action and the session is dropped.

CLI Session

Configure the registration-plan action to block.

```
NNOS-E> config vsp registration-plan
config registration-plan> config route 617
config route 617> set to-uri-match phone-prefix 617
config route 617> set action block
config route 617> return
config registration-plan>
```

Registration Throttling

In order to keep pace with the performance of the upstream registrar, you can configure the upstream peer with *registration throttling*. Registration throttling limits the registrations to the upstream server while accepting registrations from downstream clients. In this case, the OS-E delegates initial registrations to the upstream peer. Once the upstream peer grants the registration to the client with a final “200 OK” response code, the OS-E terminates subsequent re-registrations (suppressing them from delegating to the upstream peer) until the binding expiration for the upstream peer is up.

The following CLI session configures the expiration time intervals on the client and on the peer registrar. Timer settings are indicated in seconds.

CLI Session

Configure the upstream registrar.

```
NNOS-E> config vsp enterprise servers sip-gateway broadworks
Creating 'sip-gateway broadworks'
config sip-gateway broadworks> set domain as.broadworks.net
config sip-gateway broadworks> set domain as.broadworks.net
config sip-gateway broadworks> set domain-alias broadworks.net
config sip-gateway broadworks> set peer-max-interval 86400
config sip-gateway broadworks> set peer-min-interval 3600
config sip-gateway broadworks> config server-pool
```

```
config server-pool> config server primary
config server-primary> set host 12.39.208.251
config server-primary> return
config server-pool> configure server backup
config server backup> set host 12.39.208.252
config server pool> return
configure sip-gateway broadworks> return
```

Configure the registration-plan to enable registration throttling.

```
config> config vsp registration-plan
config registration-plan> config route broadworks
config route broadworks> set to-uri-match domain-suffix broadworks.net
config route broadworks> set action delegate
config route broadworks> set alter-contact local-host
config route broadworks> set peer server "vsp enterprise servers
    sip-gateway broadworks"
config route broadworks> set registration-throttling enabled
config registration-plan> return
config vsp>
```

In this configuration, when the OS-E receives a SIP REGISTER, it modifies the expiration to 86400 seconds (one day) and delegates it to BroadWorks. The BroadWorks registrar peer may respond with 7200 seconds (two hours). The OS-E then grants the client 60 seconds (if the client requests more than 60 seconds) or 30 seconds (if the client requests less than 30 seconds). The client will then register with the OS-E every 60/30 seconds. The OS-E accepts these registrations locally until the peer expiration of two hours expires.

Controlling Registration Admission Per VSP

Use the **max-number-of-registrations** property to control the number of SIP REGISTER sessions that the VSP can admit at one time. Depending on your network and the volume of SIP REGISTER requests, you can tune the setting accordingly. The default setting is 30000 concurrent SIP REGISTER sessions.

CLI Session

```
NNOS-E> config admission-control
config admission-control> set max-number-of-registrations 30000
```

Registration Multicasting

By default, the OS-E forwards REGISTERs to a single, selected server. This is how a conventional SIP proxy processes REGISTERs.

However, you can configure an upstream peer with a multicast registration pattern, where the OS-E forwards REGISTERs concurrently to all configured servers in a server pool. This allows all the servers to have the same registration data so that the OS-E load balances among the servers in the server pool. Additionally, the OS-E can easily fail over from one server to another should the primary server go down.

The OS-E performs concurrent forwarding of REGISTERs by manipulating the expiration timer, causing the phone to quickly re-register in each registration cycle. A registration cycle is registration of a binding starting from the first server to the last server. When the OS-E receives a REGISTER, it forwards it to the first server. When the OS-E receives a 200 OK from the first server, it forwards the binding to the phone, changing the expiration time to a brief interval, triggering the phone to quickly re-register with the OS-E.

When the phone re-registers, the OS-E sends the request to a second server. This continues for all configured servers.

CLI Session

Configure the upstream server.

```
NNOS-E> config vsp enterprise servers 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 registration-pattern multicast 15
config sip-gateway broadworks> config server-pool
config server-pool> config server primary
config server-primary> set host 12.39.208.251
config server-primary> return
config server-pool> config server backup
config server-backup> set host 12.39.208.252
config server-pool> return
config sip-gateway broadworks> return
config servers>
```

Configure the registration-plan.

```
NNOS-E> config vsp registration-plan
config registration-plan> config route broadworks
config route broadworks> set to-uri-match domain-suffix broadworks.net
```

```
config route broadworks> set action delegate
config route broadworks> set alter-contact local-host
config route broadworks> set peer server "vsp enterprise servers
    sip-gateway broadworks"
config registration-plan> return
config vsp>
```

This configuration enables the OS-E to multicast registrations to primary and backup registrars. In each registration cycle, the client is instructed to send the registration every 15 seconds until the last server in the pool receives the registration.

Registration Normalization

Call normalization is the process of changing all or a portion of a SIP URL so that the SIP call is routed to a particular destination. When the OS-E receives a SIP REGISTER, it first attempts to match all server peers to see if any normalization is necessary, and then matches against local settings to see if the local normalization applies.

For each server peer, if normalization setting under routing is not checked, then no normalization applies to this server. Otherwise, the OS-E matches the host portion of REGISTER against all server pool addresses, all domain-aliases and domain subnets. If there is a match, then the REGISTER host portion is normalized to the domain name configured for the server peer.

CLI Session

Configure the upstream registrar.

```
NNOS-E> config vsp enterprise servers 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 subnet-alias 12.39.208.0/24
config sip-gateway broadworks> config server-pool
config server-pool> configure server primary
config server-primary> set host 12.39.208.251
config server-primary> return
config server-pool> configure server backup
config server backup> set host 12.39.208.252
config server backup> return
config server-pool> return
config sip-gateway broadworks> return
config vsp> top
```

Location normalization does not apply if **local-normalization** is disabled or if the **local-identity** is not set. Otherwise, the OS-E matches the host portion of REGISTER against all interface addresses and all domain-aliases. If a match occurs, the SIP REGISTER host portion is normalized to the domain name configured for the VSP.

Configure the VSP.

```
NNOS-E> config vsp  
config vsp> set local-normalization enabled  
config vsp> set local-identity eng.compXYZ.com  
config vsp>
```

Configure the registration-plan.

```
config> config vsp registration-plan  
config registration-plan> config normalization broadworks  
config normalization broadworks> set match domain-suffix  
                                  broadworks.com  
config normalization broadworks> set user-normalization no
```

Viewing the Registration Routing Table

To display information about how the OS-E routes SIP call REGISTERs, use the **show registration-plan** and the **show registration-routing** commands to display the entries in the registration-plan and in the registration-routing table. The registration-plan table contains the configured entries from the vsp/registration-plan object, while the registration-routing table contains the registration lookup entries with an active registration peer. If that peer registrar becomes unavailable, the entry is removed from the registration routing table.

You execute the **show registration-plan** and the **show registration-routing** commands from the top-level OS-E prompt, or from the OS-E Management System **Status** tab. Each table displays the following information:

- plan-name
- type
- match criteria
- peer name
- action
- hits

CLI Session

```
NNOS-E> show registration-routing
plan-name      type  match                               peer-name      action  hits
-----
broadworks.com domain broadworks.com      broadworks.com accept  0
```

In this example, a call to the broadworks.com domain triggers a registration lookup that matches the registration-plan named *broadworks*.

Other show registration commands:

- **show registration-clients**
- **show registration-delegation**
- **show registration-normalization**
- **show registration-proxy**
- **show registration-service**
- **show registration-status**

For detailed information on the **show registration** commands, refer to the *Net-Net OS-E – Objects and Properties Reference*.

Configuring a PSTN Gateway

You can configure the OS-E to allow call operations if a SIP registrar is busy or down. You do this by configuring the public switched telephone network (PSTN) gateway. The PSTN gateway will change the SIP phone call to an analog phone call to a proper destination through the PSTN.

If the primary or backup becomes available while the PSTN is in use, as discovered by the latest poll, the OS-E then begins to forward REGISTERS back to the primary (or backup) SIP registrar.

The following example CLI session configures a PSTN gateway:

CLI Session

```
NNOS-E> config vsp
config vsp> config carriers
config carriers> config carrier sipura
config carrier sipura> set admin enabled
```

```
config carrier sipura> config rate-plan
config rate-plan> return
config carrier sipura> config exchange
config exchange> return
config carrier sipura> return
config carriers> return
```

```
config vsp> set pstn-gateway carrier "vsp carriers carrier sipura"
```

Chapter 6. Session Recording and Monitoring

About This Chapter

The chapter describes the configuration for recording and monitoring SIP session audio calls, IM sessions, and file transfers. By recording and monitoring SIP sessions, you can evaluate session activity to determine if additional SIP session policies should be implemented.

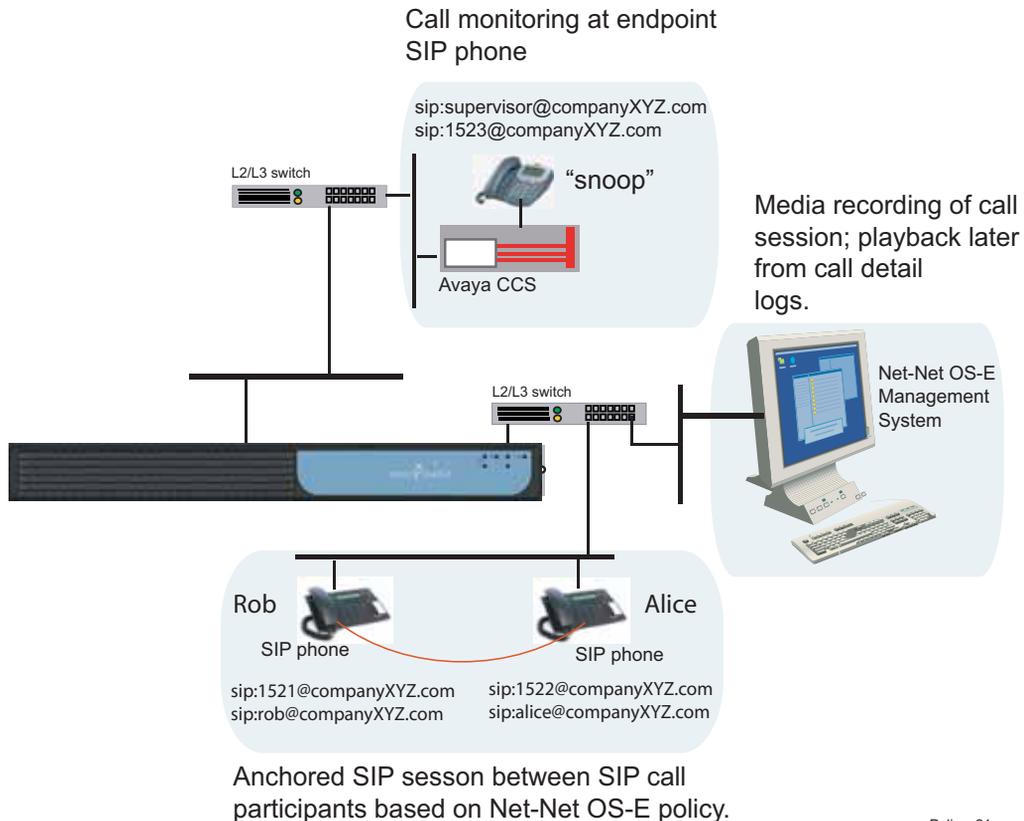
Policy-Based Media Recording and Playback

You configure media recording by creating one or more session policies under the VSP object, as described in Chapter 1, “How the OS-E Operates on SIP Sessions.”

Both media recording and monitoring are based on the specific session policy settings (rules and conditions) for matching anchored SIP session requests, with session recording and monitoring enabled.

During a SIP call session, with media recording and monitoring configured, the session is recorded locally on the OS-E system handling the call. When you configure a call monitoring policy to match the SIP session, you can forward the active SIP call to another SIP phone so that a third party can listen in on the active call. This is called *snooping*.

Once the SIP call is recorded, use the OS-E Management System **Call Logs** tab to play the recorded SIP sessions. The following image illustrates a sample network with an active session between the users *Tom* and *Alice*. The monitoring endpoint is the *supervisor@companyXYZ.com* at SIP phone extension *1523@companyXYZ.com*, who is listening to the session between Tom and Alice. The recorded session is saved in the OS-E call log where administrators can play the call.



Policy_21

Enabling Media Anchoring and Recording

Media anchoring to the OS-E system must be enabled for recording and playback of a SIP session. Anchoring forces a media stream to the OS-E, where the recording and playback mechanisms are available.

The following CLI session does the following:

- Creates a policy named *E911 Service* with an associated rule named *Record calls E911*
- Opens the **condition-list** object and sets a matching 911 condition based on the the URI regular expression
- Opens the **session-config** and **media** objects, where media anchoring and recording services are enabled
- Opens a **monitor-group** configuration where a third party can listen-in on the active call

CLI Session

```

config vsp> config policies
config policies> config session-policies
config session-policies> config policy "E911 Service"
Creating 'policy "E911 Service"'
config policy "E911 Service"> config rule "Record calls E911"
Creating 'rule "Record calls E911"'
config rule "Record calls E911"> set admin enabled
config rule "Record calls E911"> set description "E911 call policy"
config rule "Record calls E911"> config condition-list
config condition-list> set to-uri-condition user match 911
config condition-list> return

config rule "Record calls E911"> config session-config
config session-config> config media
config media> set anchor enabled
config media> config recording-policy
config recording-policy> set record enabled
config recording-policy> return
config media> return
config session-config> return
config rule "Record calls E911"> return
config policy "E911 Service"> return
config session-policies> return
config policies> return
config vsp>
    
```

```
config vsp> config monitor-group "E911 Supervisor"
Creating 'monitor-group "E911 Supervisor"'
config monitor-group "E911 Supervisor"> set admin enabled
config monitor-group "E911 Supervisor"> config monitor-endpoint
"E911"
Creating 'monitor-endpoint E911'
config monitor-endpoint E911> set to
sip:911.emergency@local.response.net
config monitor-endpoint E911> set from sip:cxc-monitor@companyXYZ.com
config monitor-endpoint E911> set transport tls
```

Configuring Call Monitor Groups and Endpoints

The call monitor-endpoint specifies the target SIP phone from which a SIP call session can be monitored while the call is in progress (called snooping). Once the SIP session is established, the OS-E rings the endpoint SIP phone where a listener can monitor the call from the beginning of the session. (There is slight time delay between actual session and the monitor endpoint.). The user at the call monitor endpoint can even terminate the RTP session between the SIP recipients, if necessary.

The following CLI session configures the monitor-group and monitor-endpoint objects:

- Creates and enables the monitor-group named “E911 supervisor.”
- Creates the monitor-endpoint named “E911 monitor” and sets the monitor endpoint phone for the active SIP call session.

CLI Session

```
config vsp> config monitor-group "E911 supervisor"
..."E911 supervisor"> set admin enabled
..."E911 supervisor"> config monitor-endpoint "E911 monitor"
..."E911 monitor"> set To sip:5121.emergency@local.response.net
config monitor-endpoint "E911 monitor">
```

Playing Back Recorded Calls

The OS-E Management System **Call Logs** tab allows you to play back previously recorded SIP calls directly on your PC. To play back calls, select **Call Logs**, and then choose the recorded call that you want to play from the list. Click **Play** to listen to the call directly from your computer.

Recording File Transfers and IM Sessions

The OS-E allows you to record copies of file transfers and IM sessions for viewing later. Just as you record SIP phone calls with the **session-config media** object, use the **file-transfer** and **instant-messaging** objects in the **session-config** to record file transfers and archive instant messaging sessions.

For File Transfer Recording

The following sample CLI session does the following:

- Creates a policy named *For file transfers* with an associated rule named *Record file transfers*
- Opens the **session-config** and **media** objects, where file transfer anchoring and recording services are enabled
- Opens the **condition-list** object and sets a matching **from-uri-condition** based on the URI regular expression, where file transfers from the URI *bob@companySierra.com* are recorded

CLI Session

```
config> config vsp policies
config policies> config session-policies
...policies> config policy "For file transfers"
..."file transfers"> config rule "Record file transfers"

...transfers"> config session-config
config session-config> config file-transfer
config file-transfer> set anchor enabled
config file-transfer> set record enabled
config file-transfer> return
config session-config> return
config rule "Record file transfers">

config rule "Record file transfers"> config condition-list
...-list> set from-uri-condition user match bob@companySierra.com
```

For Recording IM Sessions

The following sample CLI session does the following:

- Creates a policy named *For IM sessions* with an associated rule named *Archive IM sessions*
- Opens the **session-config** and **instant-messaging** objects, where instant messaging archiving services are enabled
- Opens the **condition-list** object and sets a matching **from-uri-condition** based on the URI regular expression, where IM sessions from the URI *bob@companySierra.com* are archived

CLI Session

```

config> config vsp policies
config policies> config session-policies
...policies> config policy "For IM sessions"
...IM sessions"> config rule "Archive IM sessions"

...sessions> config session-config
config session-config> config instant-messaging
config instant-messaging> set archiving enabled
config instant-messaging> return
config session-config> return
config rule "Archive IM sessions">

config rule "Archive IM sessions"> config condition-list
...-list> set from-uri-condition user match bob@companySierra.com

```

Storing Recorded Data In Specific Locations

Use the **data-locations** object under **Services** to direct recorded sessions to specific locations on the OS-E system. Recordings are classified as follows:

- **RTP-recorded**—Recording of SIP call sessions associated with a SIP audio call
- **RTP-mixed**—Playback of active SIP call sessions on target SIP phones (“snooping”)
- **file-transfer-recorded**—Recording of files that were transferred based on policy

The following CLI session shows some possible data locations (directory paths) for the three recording classifications:

CLI Session

```

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
  file-transfer-recorded[1] /cxc_common/ft_recorded
  file-transfer-recorded[2] /cxc/recorded

```

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

Simultaneous Recording To Multiple Data Locations

You can configure the OS-E to round-robin through the RTP recording directories to improve recording performance when the rtp-recorded directories point at different hard-drives. The RTP recording directories are configured with the **services/data-locations/rtp-recorded** property. Set the **services/data-locations/rtp-recorded-rotation** to **round-robin** to enable cycling through the configured directories. With the **rtp-recorded-rotation** property set to first-available, the OS-E will fill up the first volume before going to the next directory.

To record to different hard-disks when using file-mirroring, include the rtp-recorded directories in the file-mirroring directories. If the rtp-recorded directories are not included in the file-mirror directories, the data will be written to a single hard-drive.

CLI Session

```

NNOS-E config services
config services> config data-locations
config data-locations> set rtp-recorded /cxc_common/data1/rtp_rec
config data-locations> set rtp-recorded /cxc_common/data2/rtp_rec
config data-locations> set rtp-recorded-rotation round-robin
config data-locations> show

```

```

services
data-locations
  accounting-root-directory /cxc_common/accounting
  rtp-recorded[1] /cxc_common/data1/rtp_rec
  rtp-recorded[2] /cxc_common/data2/rtp_rec
  rtp-recorded-rotation round-robin
  rtp-mixed[1] /cxc_common/rtp_mixed
  file-transfer-recorded[1] /cxc_common/ft_recorded
  log /cxc_common/log

```

```
config data-locations> return
config services> return
config> config master-services
config master-services> config file-mirror
config file-mirror> set admin enabled
config file-mirror> set file-mirror-directory /cxc_common/data1/
    rtp_rec
config file-mirror> set file-mirror-directory /cxc_common/data2/
    rtp_rec
config file-mirror> show

master-services
file-mirror
  admin enabled
  host-box[1] cluster\box 1
  group 0
  file-mirror-directory[1] /cxc_common/data1/rtp_rec
  file-mirror-directory[2] /cxc_common/data2/rtp_rec

config file-mirror> return
config master-services> return
config>
```

Setting Up a Pre-Recorded Call Announcement

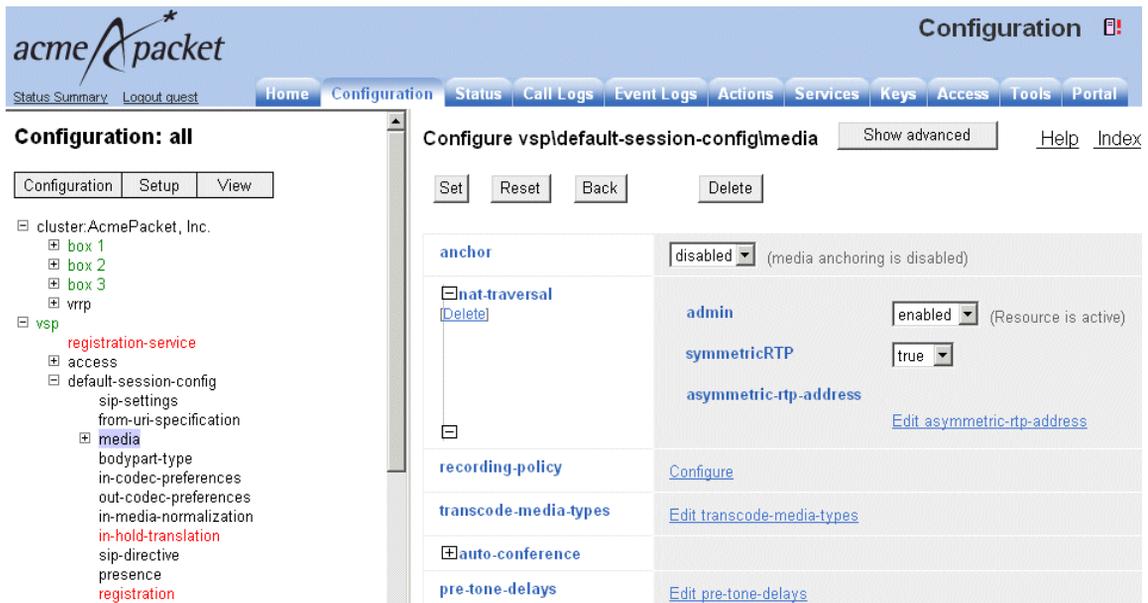
Optionally, you can record and upload a phone announcement that plays after you initiate a VoIP phone call. For each call session, the OS-E delays the received media stream until the pre-recorded message has completed. When the pre-recorded message is completed, the OS-E allows the media stream. The OS-E only plays the message in the direction of the INVITE, and media is blocked in both directions until the introduction completes.

You can create your announcement using programs such as Windows Sound Recorder (Start Menu->Programs->Accessories->Entertainment) or a shareware program such as GoldWave. Your local PC on which you are running this program must have a PC microphone in order to record the announcement to a .WAV audio file.

Once you create the .WAV recording, use the OS-E Management System **Tools** tab to upload the recorded announcement. When you initiate the VoIP call, you will hear the announcement when you pick up the phone.

Perform the following steps:

1. Using your selected sound recording program, record your message using the microphone configured on your PC to record the .WAV audio file. By default, the file should saved away in the Documents and Settings/user/Local Settings/Temp directory.
2. Using the OS-E Management System **Tools** tab, select the **Upload File** function to transfer the recorded announcement to the OS-E.
3. Open the **vsp/default-session-configuration/media** page, as illustrated in the following image.



4. Go the **Introduction** field and click **Browse System Files** to locate the .WAV file that you just recorded and uploaded to the OS-E system. Click **Set**.
5. Optionally, select **periodic-announcement** if you would like to play the pre-recorded .WAV file at intervals during the SIP call session, as illustrated in the image below.

Enter a period (in seconds) to specify the frequency at which the recording is played during the session, such as every 60 seconds. Enter a **duration** in milliseconds (1000 = 1 second) to specify how long the message is actually played. A very short duration, for example, may play only a part of your recorded message.



Note: Please ensure that the .WAV file that you uploaded is in the proper compression scheme. To test a .WAV file, upload the file to the OS-E system and use the **file-play-verify** action to check the .WAV file for compatibility. If so, the OS-E returns a “Success!” message. If the file is not compatible and is using a different compression scheme, the OS-E returns the “Unsupported Compression Code message.”

Managing Pre-Recorded Announcements For Possible Failover

In the event of a failover in a high-availability OS-E cluster, the pre-recorded file is not automatically transferred to the OS-E system that assumes mastership. As a result, the pre-recorded announcement .WAV file will not be played at the start of the next SIP call.

To enable pre-recorded announcements in high-availability networks, you will need to enable the **file-mirror master-service** to copy the announcement file to the other OS-E systems in the cluster.

Perform the following steps:

1. Enable the **file-mirror master-service** and configure a mirror directory. For example: `/cxc_common/mirror/`.
2. List all OS-E systems in the cluster as possible hosts using the **show boxes** command at the top-level CXC prompt.
3. From the OS-E Management System **Tools** tab, upload the pre-call announcement .wav file to this mirror directory. For example: `/cxc_common/mirror/announcement.wav`
4. Run the the following action:

```
NNOS-E file-mirror-service make-available /cxc_common/mirror/  
announcement.wav.
```

This can be done using the CLI or the OS-E Management System.

5. The .WAV file will now be available on every box as /cxc_common/mirror/announcement.wav.

Chapter 7. Configuring Routing Arbitration

About This Chapter

This chapter provides information on configuring the OS-E so that the least-cost routes (or preferred carriers with varying levels of service and quality metrics) are applied to a SIP call when there are multiple gateways to a SIP destination. A matching dial-plan controls the route selection over these gateways to the SIP destination.

What Is Routing Arbitration?

Routing arbitration allows different cost-based routing algorithms to be used in selecting where the OS-E forwards incoming calls. For a given destination, there may be multiple carriers that could service the call. The OS-E makes a determination where to forward the call using a routing arbitration decision based on

- The customer-preferred carrier
- Bandwidth utilization (use the most lightly used link)
- IP QoS (use the carrier link that has the best IP QoS)
- Cost (use the carrier with the lowest routing cost to that destination)

You configure routing arbitration using the VSP **dial-plan** object. The dial-plan **arbiter** determines which metrics to use in selecting a destination server. The arbiter is an ordered set of rules that configure the different cost-based routing algorithms, which the OS-E uses to select where to forward inbound SIP calls. When the OS-E receives a SIP call, it makes a determination where to forward the call (to the next hop) based on a routing arbitration decision. This is necessary because, for a given destination SIP server, multiple carriers may be available to route the call.

For complete details on the dial-plan **arbiter**, refer to the *Net-Net OS-E – Objects and Properties Reference*.

About Carriers and Gateways

The `vsp\carriers\carrier` object allows service providers to perform routing arbitration between carriers using metrics such as preferred routes, route cost, quality-of-service (QoS), and call load balancing. Each exchange and switch that you configure under the **carrier** object is an independent gateway, and each gateway along the route can be from a different carrier gateway vendor.

Additionally, a gateway is the container for carrier trunks, so that a carrier media gateway or softswitch can be properly configured. A matching **dial-plan** can point to the configuration in the following route peers:

- `carrier\carrier`
- `carriers\carrier\hunt-group`
- `carriers\carrier\exchange`
- `carriers\carrier\exchange\switch`
- `carriers\carrier\exchange\switch\trunk-group`

Note that you can configure one routing arbitration methodology a number of different ways using the **carriers** object. For example:

1. A hunt-group with one container having two gateways
2. A hunt-group having the two gateways
3. A hunt-group with two containers, each container having one gateway

Terminology

You should be familiar with the following terminology when configuring carriers.

- **Carriers**—Carriers are containers for gateways; carriers are the elements for executing routing arbitration in a network with multiple gateways to a destination.
- **Exchanges**—Exchanges are gateways in the carrier network.
- **Trunk groups**—Trunks are circuits that connect two switching systems that establishment of an end-to-end connection. Trunk groups are containers that include multiple trunks (circuits or interchangeable paths) associated with a gateway. The OS-E uses a trunk group to route calls to a PSTN gateway over specific circuits. The dial-plan **arbiter** configuration determines over which trunk group to route a call.
- **Hunt groups**—Hunt groups are containers that can include any combination of carriers, gateways, or trunks. A hunt-group comprises multiple carrier trunks that can be considered in the route selection of a call. Using dial-plan **arbiter** rule, the OS-E considers each entry in the hunt-group to calculate the most preferred carrier for a call. During routing of a call, if a failure occurs, the OS-E hunts for the next available trunk-group in the hunt-group.

How Routing Arbitration Works

When the OS-E receives a SIP INVITE, and if there is a matching **dial-plan** based on the SIP REQUEST URI, the matching dial-plan runs the following configuration objects in sequence for instructions on how to forward the call:

- Dial-prefix/normalization
- Dial-plan arbiter
- Route/source-route

If a SIP INVITE matches the dial-plan **arbiter** configuration, the arbiter uses the **vsp carriers\carrier** and the **vsp\carriers\hunt-group** configuration settings to determine the carrier, gateway, or trunk to connect the call to the destination SIP server.

By default, the dial-plan lookup uses the Request URI for dial-plan processing. However, if the INVITE is received from a server, and if the **server-pool** object (or the **vsp\carriers\carrier\gateway** object) has the **call-routing-on** property set to **to-uri**, then the dial-plan operates on the To URI portion of the SIP call.

CLI Session

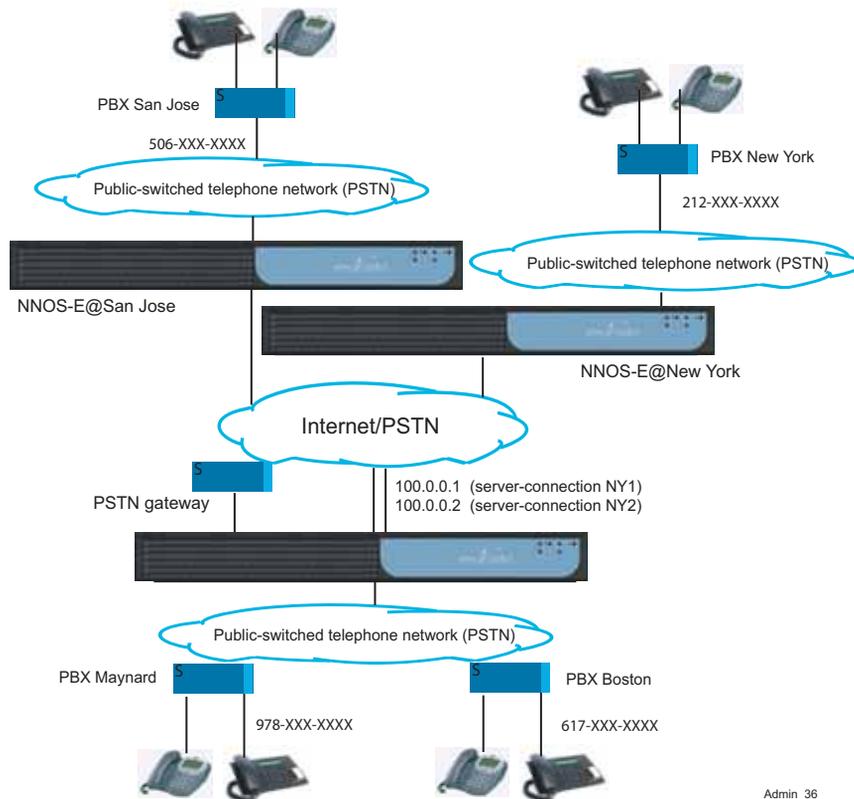
```

NNOS-E> config vsp
config vsp> config dial-plan
config dial-plan> config arbiter ABC
config arbiter ABC> set arbiter-apply best-match
config arbiter ABC> set call-hunting-type parallel
config arbiter ABC> set rule least-cost unlimited
config arbiter ABC> set rule least-calls unlimited
config arbiter ABC> set rule trunk-qos "vsp\carriers\class-of-service
high"
config arbiter ABC> set session-config-pool-entry vsp
session-config-pool entry 1

```

Sample Carrier Network

The following image illustrates a sample carrier network using three OS-E systems. A description of how to configure the network follows.



Admin_36

Configuration Steps

There are several steps that you need to perform when configuring carriers and gateways:

1. Configure the enterprise servers. These are the SIP gateways involved in the routing of SIP calls to their destination SIP servers.
2. Configure the server-pools and servers for SIP traffic sessions.
3. Configure the carrier PSTN gateway and a trunk-group where you define the rate plan that the OS-E uses in selecting the most-preferred route (one with the lowest cost) to the call destination.
4. Configure the dial-plan that arbitrates the best route to a server, carrier, gateway, trunk, or hunt-group based on matched traffic.

Configuring the Enterprise Servers (Sip Gateways)

At the local OS-E system (OS-E-Boston), you need to configure enterprise servers and PBX equipment as SIP gateways. These SIP gateways are:

- OS-E@SanJose
- OS-E@NewYork
- PBX Boston
- PBX Maynard

Each **sip-gateway** has its own **server-pool** and **server** configuration. The following image illustrates a sample management page showing the configuration associated with the SIP gateways. (Note that if you are viewing the PDF file, use the Acrobat Reader zoom function to enhance the details of this page.)

The screenshot shows the Acme Packet Configuration interface. The main content area is titled "Configure vsplenterpriseservers" and includes a "Show advanced" button and links for "Help" and "Index". Below this are buttons for "Set", "Reset", "Back", and "Delete".

The table below lists the server configurations:

	server	admin	domain	routing-tag	domain-alias	domain-subnet	ping-interval	inbound session config-pool-entry
Edit Delete	sip-gateway NNOS-E@SanJose	enabled	covergence.com				10	Edit
Edit Delete	sip-connection "PRX Boston"	enabled	covergence.com				10	Edit
Edit Delete	sip-gateway NNOS-E@NewYork	enabled	covergence.com				10	Edit
Edit Delete	h323-server 1	enabled					10	Edit
Edit Delete	sip-gateway 1	enabled					10	Edit
Edit Delete	sip-gateway PSTN	enabled					10	Edit
Edit Delete	sip-gateway broadworks	enabled					10	Edit
Edit Delete	sip-connection "PRX Maynard"	enabled					10	Edit

At the bottom of the table area, there is a link: [Add sip-connection](#)

Configuring the Server-Pools

Each SIP gateway has a **server-pool** configuration that describes the connection details and preferences to the carrier destinations.

The following image shows the server-pool configuration for the OS-E@NewYork sip-gateway. The gateway is configured with two server-connections, NY1 and NY2. Each connection has it's own unique IP address and property settings, where one connection (or route) may be preferred over another configured route.

The screenshot shows the Acme Packet Configuration interface. The main configuration area is titled "servers:" and shows a configuration for a "sip-proxy" server type. A table lists two server connections:

	server	endpoint	host	transport	port	local-ip	local-port	connection-role	connection-retry-interval
Edit Delete	server NY1	default		UDP	5060	0.0.0.0	0	initiator	5
Edit Delete	server NY2	default		UDP	5060	0.0.0.0	0	initiator	5

Below the table, there are options to "Add server" and "handle-response" with sub-options "handle-response" and "handle-pattern". There is also an "Add handle-response" link.

The left sidebar shows a configuration tree with the following structure:

- cluster:AcmePacket, Inc.
 - box 1
 - box 2
 - box 3
 - vrrp
- vsp
 - registration-service
 - access
 - default-session-config
 - autonomous-ip
 - tls
 - pre-session-config
 - policies
 - user cxc
 - static-stack-settings
 - session-config-pool
 - dial-plan
 - registration-plan
 - enterprise
 - directories

The following image shows the details for the server named *NY1*.

The screenshot displays the Acme Packet Configuration interface. The main title is "Configuration" with a sub-header "Configure vsplenterprise\servers\sip-gateway NNOS-E@NewYork\server-poolserver NY1". The interface includes a navigation menu with options like Home, Configuration, Status, Call Logs, Event Logs, Actions, Services, Keys, Access, Tools, and Portal. A left sidebar shows a tree view of the configuration hierarchy, including cluster:AcmePacket, Inc., vsp, and registration-service. The main content area shows the configuration details for the server NY1, with fields for server-name, endpoint, host, transport, port, local-ip, and local-port. The transport is set to TLS, and the certificate is vsp/tls/certificate NewYork.

Configuring SIP Connections

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

The SIP connection server type provides a client/server model between the OS-E and customer premise equipment. The OS-E fills the server role, while the connection (line) between the CPE and the OS-E 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 the OS-E.

Using this server type allows you to create a configuration specific to an AOR. For instance, it allows you to control the number of concurrent calls to (emission control) and from (admission control) the specific address-of-record (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, the OS-E 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 the OS-E 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), the OS-E can use the dynamic learn feature (if enabled), to derive the sip-connection's transport information from the client registration.

The following image shows the sip-connection page for *PBX Maynard*.



Configuring Carriers and Switches

Carriers, gateways, and exchanges support routing arbitration among multiple carrier gateways with varying levels service and quality metrics. Using a matching dial-plan **arbiter**, the OS-E uses the **carrier** configuration to select the best route for the SIP call.

The following image shows a sample carrier exchange configuration page. This page also allows you to configure **trunk-group**, also known as a group of circuits associated with a carrier switch.

The screenshot shows the AcmePacket Configuration interface. The main title is "Configure vspcarrier carrier Verizonexchange "San Jose" lswitch local". The interface includes a navigation menu with options like Home, Configuration, Status, Call Logs, Event Logs, Actions, Services, Keys, Access, Tools, and Portal. A left sidebar shows a tree view of the configuration hierarchy, including cluster, box, vrtp, vsp, registration-service, access, default-session-config, autonomous-ip, tls, pre-session-config, policies, user cxc, static-stack-settings, session-config-pool, dial-plan, registration-plan, enterprise, directories, servers, federations, 3pcc-servers, carriers, carrier-jen, carrier-1, carrier-Verizon, hunt-group-Verizon, hunt-group-1, and class-of-service-1.

The main configuration area displays the following fields:

* gateway-name	local
description	
endpoint	default (Minimum 1 characters)
host	(host name or n.n.n.n)
transport	transport <input type="button" value="UDP"/> (User Datagram Protocol)
port	5060 (at minimum 1, default=5060)
local-ip	0.0.0.0 (n.n.n.n)
local-port	0 (from 0 to 65,535)
connection-role	initiator (local initialized connection)
connection-retry-interval	5 seconds
network	Configure
handle-response	Add handle-response

Configuring Trunk Groups

The OS-E uses a trunk group, a group of trunks that connects to the same target switch or network, to route calls to a PSTN gateway over specific circuits between TDM circuit switches. It is through the dial-plan arbiter configuration that the OS-E determines over which trunk group of a particular carrier to route a call.

When the OS-E routes a call to a specific trunk, it appends a tag to indicate to the carrier over which the call should be transmitted. For example, if the OS-E routes a call to trunk "SanJose," it appends **trgp=sanjose** to the Request URI so that the provider gateway correctly transmits the call to trunk-group SanJose. (The peer trunk setting, in the dial-plan object, specifies over which trunk to route the call.)

The screenshot shows the Acme Packet configuration interface. The top navigation bar includes 'Home', 'Configuration', 'Status', 'Call Logs', 'Event Logs', 'Actions', 'Services', 'Keys', 'Access', 'Tools', and 'Portal'. The main content area is titled 'Configure vspcarriers\carrier Verizonexchange "San Jose"switch localtrunk-group 123'. Below the title are buttons for 'Set', 'Reset', 'Back', 'Copy', and 'Delete'. A message says 'Press "Set" to keep these values.' The configuration table is as follows:

* trunk-name	local
description	
* trunk-tag	123
preference	enter none or select from none (No preference applied)
handle-response	Add handle-response
admission-control	disabled (Resource is inactive)
emission-control	disabled (Resource is inactive)
max-bandwidth	enter unlimited kbits-per-second or select from unlimited
max-number-of-concurrent-calls	1500 (from 0 to 1,000,000,default=1000)
max-calls-in-setup	30 (from 0 to 10,000,default=30)
call-rate-limiting	* admin disabled
flat-rate	enter not-available cents per minute or select from not-available
external-outbound-normalization	* option no

Configuring Hunt-Groups

For each carrier, you can configure a **hunt-group** that contains a combination combination of carriers, gateways, or trunks that can be considered in the route selection of a call. Using dial-plan **arbiter** rule, the OS-E considers each entry in the hunt-group to calculate the most preferred carrier for a call.

During routing of a call, if a failure occurs, the OS-E hunts for the next available trunk-group in the hunt-group configuration. The following image shows a sample hunt-group configuration page.

The **add option** function adds entries to the hunt group. Each option is a pointer to a previously configured trunk, gateway, exchange, carrier, or enterprise server. You can add multiple options of a type, such as multiple exchange options. To add multiple types to a single option, use the pool functionality. For example, create a server configuration with a server pool that contains the servers you wish to group. You can set a preference for the entry, which the arbiter uses in its calculations. If you do not set a preference, the value is the maximum (65535). Optionally, enter a preference in the range of 0, the most preferred, to 65535. The default preference for an entry is none (equivalent to 65535), meaning that no preference is applied.

The screenshot shows the AcmePacket Configuration interface. The main configuration area is titled "Configure vspcarriers\hunt-group Verizon1". It includes a "Show advanced" button and "Help" and "Index" links. Below the title are buttons for "Set", "Reset", "Back", "Copy", and "Delete".

The configuration fields are:

- * name:** Verizon1
- admin:** enabled (Resource is active)
- option:** A table with columns for "option" and "type".

option	type
▼ Edit Delete	carrier vspcarriers\carrier Verizon none
▲ Edit Delete	carrier vspcarriers\carrier 1 0

Below the table is an "Add option" link. At the bottom of the configuration area are buttons for "Set", "Reset", "Back", and "Copy", along with "Help" and "Index" links.

Configuring the Dial-Plan Arbiter

The OS-E dial-plan **arbiter** uses the **carrier** and **hunt-group** configuration to select the carrier, switch, or trunk to connect the call to the destination SIP server. The arbiter provides set of rules that execute cost-based algorithms to select the route for SIP calls where multiple carriers are available.

If the OS-E lookup to the route arbitration table find no entries, it uses the following default settings:

- **best-match** setting for **arbiter-apply**
- Most-preferred, least-calls, and least-load routing calculation algorithms

The arbiter **subscriber-match** property defines which arbiter to apply to the SIP message. If a message matches the **subscriber-match** of an arbiter, the arbiter rules apply.

The screenshot shows the AcmePacket Configuration interface. The main content area is titled 'Configure vsp-dial-plan-arbiter Maynard'. It includes a navigation menu on the left with a tree view of the configuration hierarchy. The main configuration area is divided into two sections: 'selection:' and 'arbitration:'.

selection:

- * name:** Maynard
- admin:** enabled (Resource is active)
- description:** (empty field)
- apply-to-methods:** INVITE, REFER, MESSAGE, INFO (with Select All and Unselect All buttons)
- subscriber-match:** [Configure](#)
- condition-list:** [Delete](#)
- condition-list-match-secondary:** false

arbitration:

- arbiter-apply:** best-match
- call-hunting-type:** none (Net-Net OS-E forwards the call to a previously selected server)
- call-routing-on:** as-is (call routing decision is made on URI on request-uri or To header as specified previously)
- call-routing-lookup:** [Add call-routing-lookup](#)
- rule:**

	type
▼ Edit Delete	most-preferred
▲▼ Edit Delete	least-load
▲ Edit Delete	least-load

The **rules** property sets the criteria by which the OS-E selects the server to which it forwards calls. You can set as many rules as required for each **arbiter** object. The OS-E evaluates the rules in the order in which you create them.

For complete information on creating dial-plan arbitration rules, refer to the *Net-Net OS-E – Objects and Properties Reference Reference*.

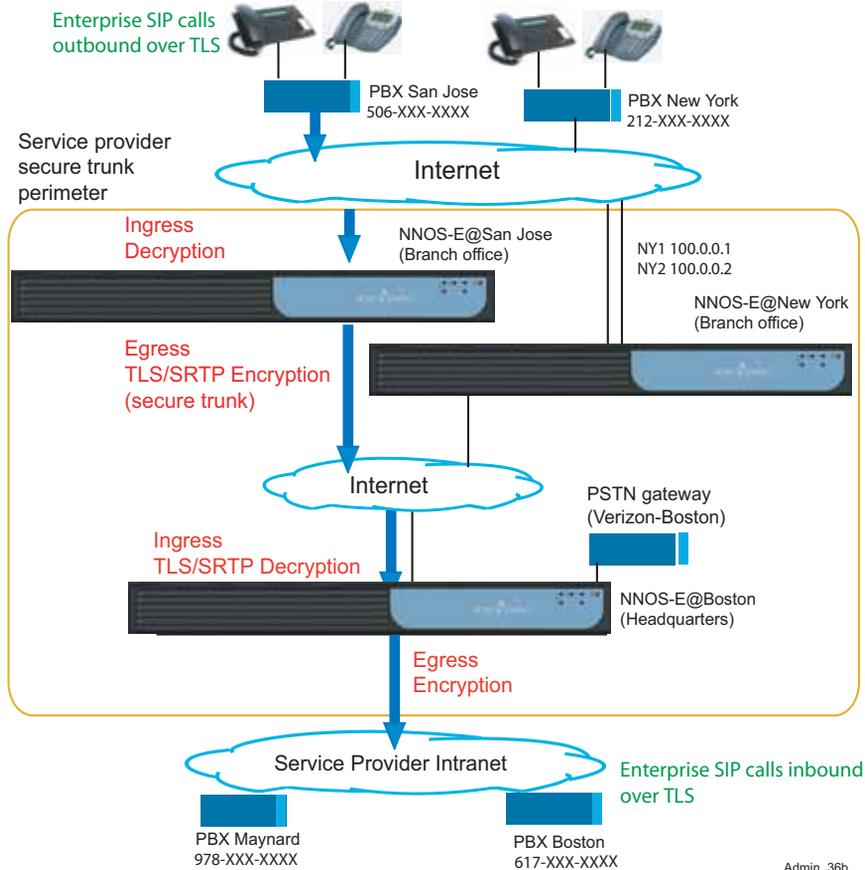
Chapter 8. Configuring Secure Trunking Networks

About This Chapter

This chapter provides information on configuring secure network trunks. A secure trunk uses TLS on the signaling stream and SRTP on the media stream between the OS-E systems in enterprise and service provider networks.

Sample Secure Trunking Networks

The following image illustrates a sample service provider network using the OS-E systems in three branch offices. Traffic between each OS-E node uses TLS and SRTP encryption on the signaling and media streams. It assumes that the phones in this network do not support SRTP. (See the note at the end of the next section.)



Call Traversal In the Secure Trunk

Starting at the top of the illustration in the above image:

1. SIP phone calls over a TLS transport from subscribers enter the secure trunk at the OS-E@San Jose, where the inbound call signaling stream is decrypted using the OS-E's inbound session configuration.
2. Using the outbound session configuration, the OS-E@San Jose then reencrypts the TLS signaling stream, and then encrypts the SIP media stream using SRTP. This creates the secure trunk as the SIP call session traverses the Internet to the call destination at OS-E@Boston.
3. When the SIP call reaches the OS-E@Boston, the inbound session configuration decrypts the TLS and SRTP call streams.
4. The outbound session configuration at OS-E@Boston then reencrypts the SIP signaling stream to TLS before forwarding the call to the destination.



Note: Currently, most SIP phones do not support SRTP. Therefore, media streams outside of the secure trunk are sent “in the clear” to the call destination when received by the service provider. However, you can configure the outbound session configuration to perform SRTP encryption and offer it the destination phone. If the phone does not support the SRTP, then RTP is used to deliver the media stream.

Configuration Steps

There are several steps that you need to perform to configure a secure trunking network.

1. Configure the enterprise servers and connections. These are the SIP gateways involved in the routing of SIP calls to their destination SIP servers.
2. Configure the server-pools.
3. Configure the SIP connections — the SIP PBXs communicating with the OS-E you are configuring.)
4. Configure dial-plan routes to SIP gateways and connections.

5. Configure the inbound and outbound session configuration entries for the encryption and decryption policies, and then apply the entries to the SIP gateways and SIP connections. There are the ingress and egress points for encryption and decryption.
6. Configure the carriers and trunk groups.

Net-Net OS-E Encryption/Decryption Policies

For SIP gateways and SIP connections, you need to define encryption and decryption policies to the inbound (ingress) and outbound (egress) SIP sessions. The following table lists the encryption methods used on the ingress and egress sessions and the encryption or decryption policy applied to each session. In a secure trunk, the SIP media and signaling streams are encrypted using TLS (on signaling) and SRTP (on media).

Server type	Encryption methods	Encryption/decryption policy (default names)
SIP gateway	IN-ENCRYPTION (on OS-E ingress call direction)	NNOS-EDecryptPolicy
SIP gateway	OUT-ENCRYPTION (on OS-E egress call direction)	NNOS-EEncryptPolicy
SIP connection	IN-ENCRYPTION (on OS-E ingress call direction)	GenericDecryptPolicy
SIP connection	OUT-ENCRYPTION (on OS-E egress call direction)	GenericEncryptPolicy

How the OS-E Performs Encryption and Decryption

Encryption is symmetric on inbound and outbound sides of the OS-E, regardless of the call direction. This means that encryption and decryption always occurs to that side, or it does not. This is determined at call setup time.

In the following image, the red arrows on the left indicate encrypted media, and blue arrows on the right indicate plaintext media. The OS-E performs the encryption and decryption to make the SIP session either encrypted or plaintext.



The **in-encryption** policy causes the left side to be encrypted and decrypted. The SIP phone call initiated on the left must propose encryption in the Session Description Protocol (SDP) as well as provide the encryption key. If the in-encryption mode is set to require and the phone does not send it, the OS-E rejects the call. The in-encryption mode must be set to allow or require, as proposed by the SIP phone.

The **out-encryption** policy determines whether we do encryption/decryption on the right side. If the out-encryption mode is set to offer or require, the OS-E proposes encryption in the SDP sent to the right side. If the right side does not want encryption, the destination phone may send the OS-E an SDP that does not contain encryption, therefore rejecting the call. If the OS-E is set to require encryption and the received SDP has no encryption, the OS-E rejects the call. If the OS-E is set to offer encryption and the received SDP has no encryption, the OS-E allows the call to proceed without encryption or decryption on the right side, as shown in the above image.

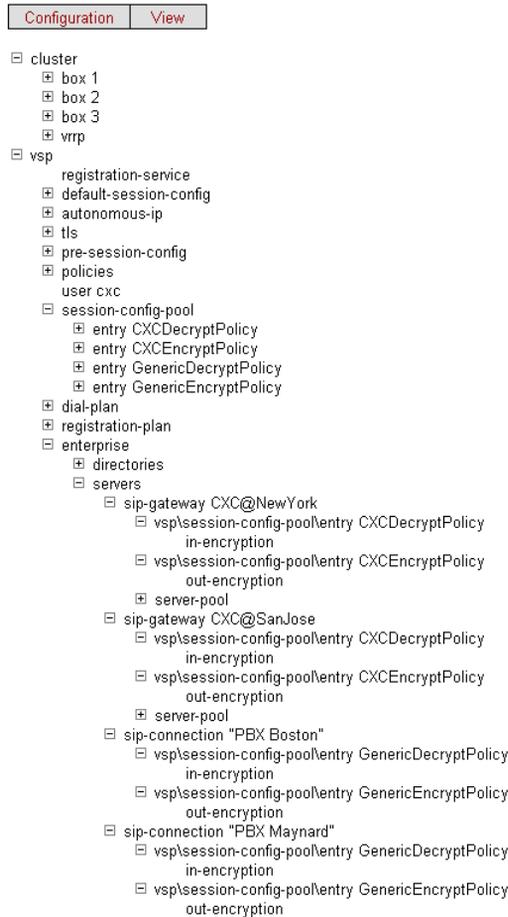
Applying Encryption and Decryption Policies

You configure encryption and decryption modes when you configure SIP gateways and SIP connections. In the OS-E Management System, go to the Policies section on the `vsp/enterprise/servers/sip-gateway` or `vsp/enterprise/servers/sip-connection` page and specify the **inbound-session-config-entry** and **outbound-session-config entry** by making a selection from the session-config-pool.

Open the session-config-pool and create entries for the in-encryption and out-encryption properties. Specify the encryption mode and encryption type as required for the ingress and egress call streams. In-encryption modes include: **disable**, **allow**, and **require**. Out-encryption modes include: **none**, **offer**, **require**, and **follow**.

The following image illustrates the configuration tree showing the encryption and decryption policies and where they are applied under each SIP gateway and SIP connection.

Configuration: all

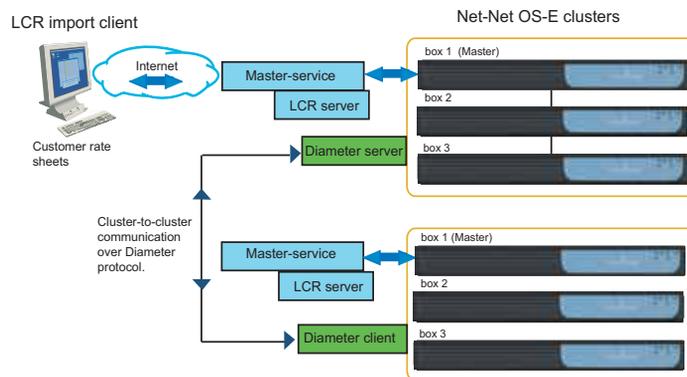


Chapter 9. Configuring Route Server Services

About This Chapter

The route server is a client-server application that provides preferred SIP call routing information to Oracle OS-E network clusters. Using an embedded route server import client (containing call rate data), the Diameter protocol, and the route server master-service running under the OS-E (the route server), network clusters can request the latest and least-cost route definitions for forwarding calls from SIP phone clients.

The following image illustrates a sample route server network showing the route server import client, the route server (running as a master-service on the OS-E), and the Diameter protocol running between two OS-E clusters for route sharing and distribution.



The route server import client is an embedded Web application on the OS-E. It has a PostgreSQL database server, an embedded Tomcat server, and a Java-based Web application. Customer rate sheets, available in a variety of formats and normalized by the import client in standard XML format, are uploaded to the route server and the route server route database.

You configure the route server as a **master-service** object on the running OS-E. Via the web-services that is running on the same OS-E as the route server, the route server receives the rate sheet updates and the latest route definitions from the import client. Using the Diameter protocol, the route server communicates route definitions with other OS-E systems within a cluster and among other network clusters.

Note: Oracle recommends configuring the route server and route server import client on two separate OS-Es with Web Services configured on the route server OS-E.

The OS-E makes requests for the least cost routes from the route server by sending the call To: and From: URLs to the route server. The route server then looks up the least cost routes in its database using the To: and From: URLs and returns a list of routes available to the OS-E, with the list sorted by priority and route cost.

After receiving the list of routes from the route server, the OS-E compares the routes with any configured routes before applying filtering criteria, such as MOS and load restrictions. The OS-E attempts all routes until a successful response is received from the call destination.

Note: When upgrading a version of the OS-E where **lcr-import-service** is running, ensure the **lcr-import-service** object is disabled before installing a new supertar.

Downloading and Installing the Route Server Import Client

Oracle provides the route server import client with the download named **emblcrimport_release#_build#.tar.gz**. The installed client communicates with the route server over an HTTPS Web connection.

Before Installing the Route Server Import Client

To view and use the route server import tool, the user must have **lcr-import** permissions granted. Users can be granted one of the following route server permissions:

- **enabled**—Allows the user to perform all route server functions.
- **view**—Allows the user to perform read-only tasks.
- **disabled**—Bars the user from all access to the route server import tool.

For more information on configuring user permissions, see Management System Access in this guide.

Before installing the route server import tool, you must have the OS-E software installed and running properly. Also, you must configure the following:

- Enable the **master-services > database** object.
- Add a certificate via the **vsp > tls** object if you are using certificate-based authentication. This step must be executed on both the OS-E running the import tool and the OS-E running the route server.
- Enable the **web-service** object (on the OS-E where the route server is running) and select the type of authentication to use for communication between the web-service and route server import tool. Note web-services does not have to be enabled on the OS-E with the import client installed.

Note: The route server import tool is not supported in a cluster environment.

Enable the master-services database

To use the route server import tool properly, the **master-services > database** object must be **enabled**. If you attempt to use the route server import without the database enabled, you get the following error message.

```
"Cannot fetch the LCR import records. [internal error: No suitable driver found for jdbc:postgresql://127.0.0.1:5432/lcrimport]"
```

1. Log in to the OS-E.
2. Under the **Services** tab, enable the **master-services > database** object.

The screenshot shows the Acme Packet web interface. The top navigation bar includes 'Home', 'Configuration', 'Status', 'Call Logs', 'Event Logs', 'Actions', 'Services', 'Keys', 'Access', and 'Tools'. The 'Services' tab is selected. On the left, a sidebar shows a tree view with 'services' expanded to 'master-services' and 'database' selected. The main content area is titled 'Configure master-services/database' and includes a 'Show advanced' button and links for 'Help' and 'Index'. Below this are buttons for 'Set', 'Reset', 'Back', and 'Delete'. The configuration table is as follows:

admin	enabled (Resource is active)
host-box	clusterbox 1 Edit host-box
group	0 (from 0 to 32, default=0)
maintenance	type: time-of-day time: 03:00:00.00000
media	disabled (Resource is inactive)

At the bottom of the configuration area are buttons for 'Set', 'Reset', and 'Back', along with 'Help' and 'Index' links.

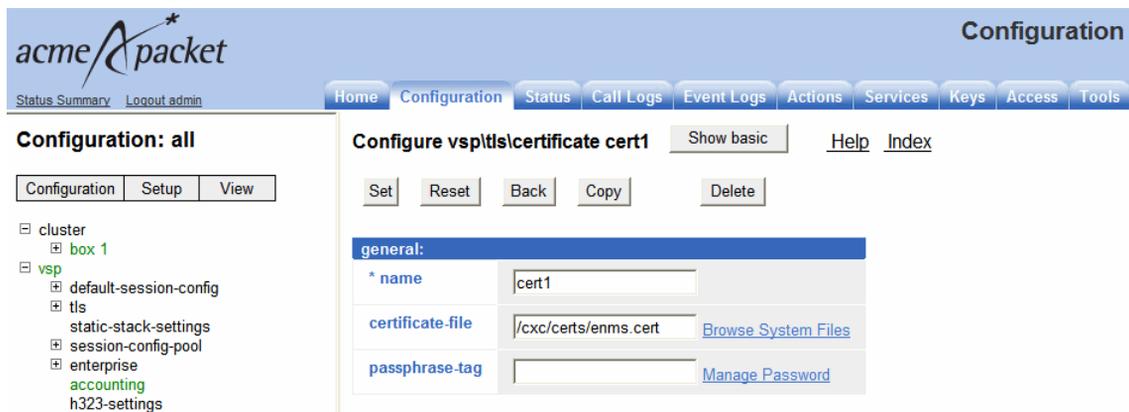
3. Update and save the configuration.

Adding a certificate to the TLS object

You must configure a TLS entry to add the appropriate certificate when using certificate-based authentication.

1. Under the **Configuration** tab select **tls**, then click **Add certificate**.
2. Enter the name you are using for the certificate and click **Create**.

- Browse to the certificate file in the **certificate-file** property.



- Browse to the **lcr-import-service** object and under **Authentication**, select the certificate.
- Click **Set**.
- Update and save the configuration.

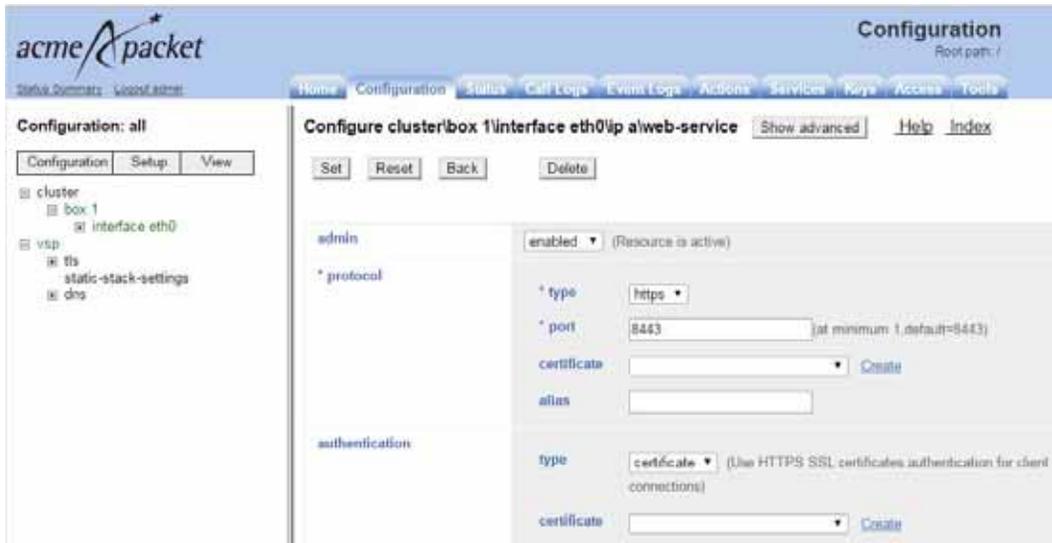
Enabling and Updating Web-Services On the Route Server

One the OS-E running the route server, you must also enable the **web-service** object and select the type of authentication to use. The Web Services communicates with the import tool and invokes route server commands on behalf of the authenticated import client.

- Under the **Configuration** tab, use the following path to get to the web-service object: **cluster > box > interface > ip > web-service**.
- admin**—Set to **enabled**.
- protocol**—Select the protocol **type** and **port** to use for communication between the web-service and the route server import tool. The default is **http 8080**. The default **https** port is **8443**.
- authentication**—Specify whether you want to use basic or certificate-based authentication. The default is **basic**.

Note: Oracle recommends using the HTTPS protocol and certificate authentication.

When set to **certificate**, specify the certificate you want to use.



5. Update and save your configuration.

Downloading and Installing the Route Server Import Tool

You download the route server import tool as an embedded application on the OS-E.

The route server import tool must be run on an OS-E system dedicated just for this purpose. You cannot use this OS-E for anything else.

Note: To run the route server import tool, the DOS process must be running at level 7. To ensure you have this version running, execute the **show processes** action under **System** on the left hand list below the **Status** tab.

6. Upload the **emblcrimport_release#_build#.tar.gz** file to the “/releases” directory on your OS-E using either the **Upload file** action under the **Tools** tab.

The screenshot shows the acmeApacket web interface. The top navigation bar includes 'Home', 'Configuration', 'Status', 'Call Logs', 'Event Logs', 'Actions', 'Services', 'Keys', 'Access', and 'Tools'. The left sidebar lists various tool actions: 'Update software', 'Retrieve license', 'Upload license file', 'Upload file', 'Download file', 'Download saved configuration file', and 'Compare configuration files'. The main content area is titled 'Upload File To Net-Net OS-E' and contains the following text: 'You can upload a file from your computer to Net-Net OS-E. You can keep the original name or specify a new name for the file on the box.' Below this text are form fields: 'File:' with the value '\\Acmeland2\acmeland1\ACME Technical Publ' and a 'Browse...' button; 'Destination Path:' with the value '/cxc_common/releases'; 'Overwrite Existing File:' with an unchecked checkbox; and 'Destination Name:' with an empty text box. An 'Update' button is located at the bottom right of the form.

Or the **Update software** action under the **Tools** tab.

The screenshot shows the acmeApacket web interface. The top navigation bar includes 'Home', 'Configuration', 'Status', 'Call Logs', 'Event Logs', 'Actions', 'Services', 'Keys', 'Access', and 'Tools'. The left sidebar lists various tool actions: 'Update software', 'Retrieve license', 'Upload license file', 'Upload file', 'Download file', 'Download saved configuration file', and 'Compare configuration files'. The main content area is titled 'Update Software' and contains the following text: 'Please enter the path to the software update to be copied to Net-Net OS-E. To install the software, check the install option. Then, press Update.' Below this text are form fields: 'Software Update File:' with an empty text box and a 'Browse...' button; 'Install The Update?' with an unchecked checkbox; and 'Realm' with radio button options for 'box', 'cluster', and 'controlled'. An 'Update' button is located at the bottom right of the form.

- Under the **Actions** tab, execute the **install file releases embldcrimport_release#_build#.tar.gz** action to install the route server import tool to the OS-E.

The screenshot shows the 'acme packet' web interface. The left sidebar lists various actions, with 'install' selected. The main content area is titled 'install' and 'install a software upgrade'. It contains a configuration form with the following fields:

- * data**: (empty)
- * mode**: dropdown menu set to 'file' (install the specified upgrade)
- * source**: text input field containing '/cxc_common/releases/nr' with a 'Browse System Files' link.
- realm**: dropdown menu set to '<Not configured>'

At the bottom of the form are 'Invoke' and 'Schedule' buttons.

- The OS-E restarts.

Configuring the Route Server Import Tool

This section explains how to configure the route server import tool.

- Under the **Configuration** tab, select the **ip** object under the **cluster > box > interface** object.
- Click **Configure** next to **lcr-import-service**.

The screenshot shows the 'acme packet' web interface. The left sidebar shows the configuration tree under 'Configuration: all'. The tree is expanded to show 'cluster > box 1 > interface eth0 > ip a'. The right sidebar shows a list of services with 'lcr-import-service' selected. The 'lcr-import-service' row has a 'Configure' link.

Service	Action
web-service	Configure
eventpush-service	Configure
lcr-import-service	Configure
ipsec-tunnel	Add ipsec-tunnel
ipsec-transport	Add ipsec-transport
ike	Configure
sip	Configure

3. Select the protocol **type**, **http** or **https**, and specify the target **port**. The default is **http 8082** and the default **https** port is **443**. For security purposes, Oracle recommends using secure HTTPS both here and on the route server **web-services**.

The screenshot shows the Acme Packet configuration web interface. The top navigation bar includes 'Home', 'Configuration', 'Status', 'Call Logs', 'Event Logs', 'Actions', 'Services', 'Keys', 'Access', 'Tools', and 'Portals'. The main content area is titled 'Configuration: all' and shows a tree view on the left with 'cluster' expanded to 'box 1', then 'interface eth0', and 'ip a'. The main panel is titled 'Create clusterbox 1interface eth0lcr-import-service - Step 1 of 1: Edit lcr-import-service' and contains a form for configuring the service. The form fields are: '* protocol' (set to 'https'), '* type' (set to 'https'), '* port' (set to '8082'), 'redirect-port' (set to '0'), 'certificate' (a dropdown menu), and 'alias' (an empty text field). Below the form are 'Create', 'Reset', and 'Cancel' buttons.

4. Click **Create**.

- Configure the route server import values the way you want to implement the functionality. For more information on the **lcr-import-service** properties, see the *Oracle Communications OS-E Objects and Properties Reference Guide*.

The screenshot shows the Acme Packet Configuration GUI. The main configuration area is titled "Configure cluster/box 1/interface eth0/ip a/lcr-import-service". It includes a navigation menu at the top with options like Home, Configuration, Status, Call Logs, Event Logs, Actions, Services, Keys, Access, and Tools. On the left, there is a configuration tree showing a hierarchy of objects: cluster, box 1, interface eth0, ip a, vsp, tis, static-stack-settings, and dns. The main configuration area contains several sections:

- admin**: A dropdown menu set to "enabled" with the note "(Resource is active)".
- * protocol**: A dropdown menu set to "https".
- * port**: A text input field containing "8787" with a note "(at minimum 1, default=443)".
- redirect-port**: A text input field containing "0" with a note "(from 0 to 65,535)".
- certificate**: A dropdown menu with a "Create" link.
- alias**: A text input field.
- authentication**: A dropdown menu set to "certificate" with a note "(Use custom HTTPS SSL certificates for LCR Import Service)".
- certificate**: A dropdown menu with a "Create" link.
- max-threads**: A text input field containing "10" with a note "(from 1 to 50, default=10)".
- min-spare-threads**: A text input field containing "1" with a note "(from 0 to 50, default=1)".
- max-spare-threads**: A text input field containing "5" with a note "(from 0 to 50, default=5)".
- idle-timeout**: A text input field containing "30" with a note "minutes".
- clphers**: A text input field.
- use-https-for-file-copy**: A dropdown menu set to "enabled" with the note "(Resource is active)".

At the bottom of the configuration area, there are buttons for "Set", "Reset", and "Back".

- Click **Set**. Update and save your configuration.

The route server import process must be running at Level 7.

To view the route server import process status:

- Select the **Status** tab and expand **System** on the list at the left hand of the screen.

2. Select processes.

The screenshot shows the 'acmePacket' web interface. The main content area is titled 'processes - process status'. Below the title, there is a 'View' dropdown set to 'Basic' and a 'Search' input field. To the right of the search field is a 'seconds' input field and a 'Refresh' button. The main part of the page is a table listing various system processes.

process	id	condition	run-level	starts	uptime	fds
monitor	14187	running	7	1	0 days 17:35:16	17
rmanager	14391	running	7	1	0 days 17:35:16	38
SIP	14468	running	7	1	0 days 17:35:00	43
media	14469	running	7	1	0 days 17:35:00	24
auth	0	idle	init	0	0 days 00:00:00	0
reg	14470	running	7	1	0 days 17:35:00	16
H323	0	idle	init	0	0 days 00:00:00	0
dir	0	idle	init	0	0 days 00:00:00	0
web	14438	running	7	1	0 days 17:35:13	163
WS	0	idle	init	0	0 days 00:00:00	0
acct	0	idle	init	0	0 days 00:00:00	0
dos	0	idle	init	0	0 days 00:00:00	0
SSH	14437	running	none	1	0 days 17:35:13	4
LCR	0	idle	init	0	0 days 00:00:00	0
sampling	0	idle	init	0	0 days 00:00:00	0
userdb	0	idle	init	0	0 days 00:00:00	0
presence	0	idle	init	0	0 days 00:00:00	0
eventpush	0	idle	init	0	0 days 00:00:00	0
LCRImport	14608	running	7	1	0 days 17:33:07	164
archiver	0	idle	init	0	0 days 00:00:00	0

At the bottom of the page, it says 'Page 1 of 1 showing 25 items' and 'Taken Mar 16, 2011 9:57:18 AM'.

You must now create a file event log for route server import functionality.

1. Select the **Services** tab.
2. Select **event-log** and click **Add file**.
3. Name the file and click **Create**.

- Configure the event log properties for the route server import functionality. For more information on the **services > event-log > file** properties, see the *Oracle Communications OS-E Objects and Properties Reference Guide*.

The screenshot shows the acme packet web interface. The top navigation bar includes links for Home, Configuration, Status, Call Logs, Event Logs, Actions, and Services. The main content area is titled 'Configure servicesevent-logfile lcrimport' and includes a 'Set' button, a 'Reset' button, a 'Back' button, a 'Copy' button, and a 'Delete' button. The configuration form has the following fields:

- file**:
- admin**: (Resource is active)
- filter**: [Add filter](#)
- size**: Mbytes(from 1 to 100,default=10)
- count**: (from 1 to 20,default=5)

At the bottom of the form, there are buttons for 'Set', 'Reset', 'Back', and 'Copy', along with 'Help' and 'Index' links.

- Click **Set**. Update and save your configuration.

To launch the route server import tool:

- Open a browser window and entering either:
 - `http://<ipNumber>:<port>/lcrimport`
 - `https://<ipNumber>:<port>/lcrimport`

Use either HTTP or HTTPS depending on the configured **lcr-import-service** protocol type. For security purposes, Oracle recommends using secure HTTPS.

The `<ipNumber>` value is the route server's management IP address and `<port>` is the value specified in the **lcr-import-service > protocol type**.

Note: Prior to installing the OS-E supertar, ensure the **lcr-import-service** object is disabled.

2. You can now log into the route server import login page.

Acme Packet Net-Net OS-E LCR Import Login

Name:

Password:

HTTPS Support for Call Rate Files Transferring

By default, the route server import tool uses HTTPS to transfer call rate files between it and the client server. HTTPS is a combination of HTTP and SSL/TLS protocols which provides encryption and secure identification of the server.

You can use a custom certificate for authentication by adding a custom certificate under the **vsp > tls** config object. For more information, see [Adding a certificate to the TLS object \(page 9-124\)](#).

The route server import tool only supports custom certificates created with the “PKCS12” format. If you do not specify a custom certificate and certificate alias, the import tool uses the default certificate. For more information on configuring a “PKCS12” key store, see Chapter 5, “Installing Certificates and Commissioning TLS Networks” in the *Oracle Communications Application Session Controller Installation Guide*.

When you select not to use SCP for file copy, the **lcr-import-service > use-https-for-file-copy** property takes effect. This property specifies whether to use HTTP or HTTPS to copy call rate files between the route server import tool and the route server. By default this property is enabled and uses HTTPS. When disabled, the route server import tool uses HTTP.

When you configure a pre-3.6m4 route server, HTTPS is not supported and you must use either HTTP or SCP to copy files to the route server.

To use HTTPS to transfer call rates files to the route server:

1. Log into the OS-E management interface and browse to the **lcr-import-service** configuration object.
2. Set **protocol** to **https**.

3. Set the **port** number to use to communicate with the route server web services.
4. Select the **authentication** you are using.
5. Set the **ciphers** to **TLS_RSA_WITH_AES_128_CBC_SHA**, **TLS_RSA_WITH_AES_256_CBC_SHA**. In many web browsers this can prevent weak SSL errors.
6. Set **use-https-for-copy-file** to **enabled**. This configures the OS-E to use HTTPS to transfer call rates files.

The screenshot shows the Acme Packet Configuration interface. The main content area is titled "Configure cluster/box 1/interface eth0/ip a/lcr-import-service". The configuration is as follows:

- admin:** enabled (Resource is active)
- protocol:**
 - type: https
 - port: 8787 (at minimum 1, default=443)
 - redirect-port: 0 (from 0 to 65,536)
 - certificate: [dropdown menu] Create
 - alias: [text input]
- authentication:**
 - type: certificate (Use custom HTTPS SSL certificates for LCR Import Service)
 - certificate: [dropdown menu] Create
- max-threads:** 10 (from 1 to 50, default=10)
- min-spare-threads:** 1 (from 0 to 50, default=1)
- max-spare-threads:** 5 (from 0 to 50, default=5)
- idle-timeout:** 30 minutes
- ciphers:** [text input]
- use-https-for-file-copy:** enabled (Resource is active)

Buttons for "Set", "Reset", "Back", and "Delete" are visible at the top and bottom of the configuration area.

7. Save and update your configuration.
8. Launch and log into the route server import tool.
9. Under the **Route Server** tab set **Route Server IP** to the address of the OS-E interface.

10. Set the route server **Port** to the port of the OS-E **web-services** and specify the following:
 - Web services protocol
 - Remote Authentication
 - Updating using SCP: No
11. Set **Update using SCP** to **No**. When this property is set to **No**, the route server import tool uses the value configured in the **use-https-for-file-copy**.

acme packet

Status Rates DID Ranges Route Server Audit Log

Route Server

Route Server IP:

Route Server Port:

Web service protocol:

Remote Authentication:

Update using SCP:

Activate routes on route server?

Use default route filename?

To use HTTP to download call rates files to the route server:

Note: For security reasons, Oracle does not recommend using HTTP to download call rate files to the route server.

1. Under the **lcr-import-service** configuration object, set **protocol** to **https**.
2. Set the **port** number you are using to access the route server import tool web interface.

- Set **use-https-for-file-copy** to **disabled**. This configures the OS-E to use HTTP to download call rates files.

The screenshot shows the Acme Packet configuration interface. The main title is "Configure cluster/box 1/interface eth0/ip a/lcr-import-service". The interface includes a navigation menu at the top with options like Home, Configuration, Status, Call Logs, Event Logs, Actions, Services, Keys, Access, and Tools. On the left, there is a "Configuration: all" sidebar with a tree view showing the current configuration path: cluster > box 1 > interface eth0 > ip a. The main configuration area has several sections:

- admin**: A dropdown menu is set to "enabled" (Resource is active).
- * protocol**: A dropdown menu is set to "https".
- * port**: A text input field contains "8787" (at minimum: 1, default=443).
- redirect-port**: A text input field contains "0" (from 0 to 65,536).
- certificate**: A dropdown menu with a "Create" link.
- alias**: A text input field.
- authentication**: A dropdown menu is set to "certificate" (Use custom HTTPS-SSL certificates for LCR Import Service).
- certificate**: A dropdown menu with a "Create" link.
- max-threads**: A text input field contains "10" (from 1 to 50, default=10).
- min-spare-threads**: A text input field contains "1" (from 0 to 50, default=1).
- max-spare-threads**: A text input field contains "5" (from 0 to 50, default=5).
- idle-timeout**: A text input field contains "30" minutes.
- ciphers**: A text input field.
- use-https-for-file-copy**: A dropdown menu is set to "disabled" (Resource is inactive).

At the bottom of the configuration area, there are "Set", "Reset", and "Back" buttons.

- Update and save your configuration.
- Log into the route server import tool.
- Under the **Route Server** tab set the **Route Server IP** to the address of the OS-E **web-service** interface.
- Set the route server **Port** to the port of the OS-E **web-services**.

- Set **Update using SCP** to **No**. When this property is set to **No**, the route server import tool uses the value configured in the **use-https-for-file-copy**

The screenshot shows the 'Route Server' configuration page. At the top, there is a navigation bar with tabs for 'Status', 'Rates', 'DID Ranges', 'Route Server', and 'Audit Log'. The 'Route Server' tab is active. Below the navigation bar, the 'Route Server' section contains several configuration fields: 'Route Server IP' with the value '172.30.80.29', 'Route Server Port' with '8080', 'Web service protocol' set to 'https', and 'Remote Authentication' set to 'certificate'. The 'Update using SCP' dropdown menu is set to 'No'. There are two checkboxes: 'Activate routes on route server?' which is checked, and 'Use default route filename?' which is unchecked. At the bottom of the form, there are two buttons: 'Update routes' and 'Get active routes'.

For information on configuring SCP to download call rates files to the route server, see the Oracle Communications OS-E Session Services Guide.

Using Variables in the Route Server Import Tool

The route server import tool includes a variables field on the Rates and DID Ranges pages. When these values are included, the generated routes in the route files include a variables property.

To add variables to a route file, specify name and value pairs in the CSV file in the following format:

```
name1=value1;name2=value2
```

Separate multiple name and value pairs with semi-colons “;”. During Rates and DIDs imports, specify the CSV variables field in the variables property in the import tool.

NOTE: Variable values supercede the CSV file field value when both are specified.

Using Different Rate Sheet Formats

The route server import client accepts multiple customer rate sheet formats. The rate sheet should be a CSV (comma separated value) file that lists at least three properties:

- The dialed NPA/NXX prefix or the LATA code
- End point server
- Rate

The rate sheet is normalized by the import client into a standard XML file. The file is uploaded to the route server (a master-service on the OS-E) which is then loaded into the route server database.

The format of the XML file is as follows:

```
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="mgmt_data.xsd">
  <ExternalCarrierRouteEntry index="1531207">
    <toURLMatch>212709</toURLMatch>
    <fromURLMatch>.*</fromURLMatch><carrier>bozol23</carrier>
    <emergency>true</emergency>
    <priority>33</priority>
    <rate>0.0086</rate>
    <effectStartDate>00:00:00 2008-08-08 </effectStartDate>
    <effectEndDate>00:00:00 2009-08-08</effectEndDate>
    <minDuration>44</minDuration>
    <billIncrement>55</billIncrement>
    <aniAlteration>sip:username1@domain </aniAlteration>
    <dnisAlteration>sip:username2@domain </dnisAlteration>
    <requestAlteration>sip:username3@domain </requestAlteration>
    <pAssertAlteration>sip:username4@domain </pAssertAlteration>
    <validTimeDays><days><mon>true</mon><wed>true</wed><fri>true</fri></
      days></validTimeDays>
    <sessionConfig>sessCfg123</sessionConfig>
  </ExternalCarrierRouteEntry>
```

Configuring the Route Server

The route server is an application on the OS-E that includes the following services:

- The route server process, configured as a **master-service**;
- The Diameter client and server as transport agents of the request and response messages between the route server and route server service; and
- The route server service that resides on each OS-E SIP process.

There are several configuration steps that you need to perform.

On the OS-E Running the Route Server:

1. Enable Web Services.

On the OS-E Master In the Cluster:

1. Enable the route server.
2. Configure the Diameter protocol port.

On All Other OS-E Systems In the Cluster:

1. Define a Diameter group under the VSP configuration;
2. Configure the route server service
3. Configure an enterprise server or carrier under the VSP configuration
4. Configure route arbitration

Enabling Route Server on the OS-E master

The following CLI session configures route server on the OS-E master.

CLI Session

```
NNOS-E> config master-services
config master-services> config route-server
config least-cost-routing> set admin enabled
config least-cost-routing> set host-box "cluster box 1"
config least-cost-routing> set max-routes automatic
```

When route server is enabled on multiple OS-E systems in a cluster, any rate table updates taking place on the route server master are propagated to the backup platforms. Therefore, file mirroring is not required.

The secondary property called **max-routes**, which limits the maximum allowed route entries in the rate table to be imported into the route server. Because this is a licensed feature, a license is required from Oracle in order to change the default setting (automatic).

Configuring Diameter Servers

The Diameter protocol is responsible for both inter- and intra-cluster communication of route server call rates. The following session configures a Diameter server protocol port on the OS-E.

For complete information on the Diameter protocol settings, refer to the Oracle Communications OS-E Objects and Properties Reference Guide.

CLI Session

```
config> config cluster box 1
config box 1> config interface eth1
config interface eth1> config ip a
Creating 'ip a'
config ip a> config diameter
config diameter> config origin-host east.ocl.com
config diameter> config origin-realm ocl.com
config diameter> config port 3686
Creating 'port 3686'
config port 3686> set admin enabled
config port 3686> set application routing
config port 3686> set peer-access-control none
```

Defining Diameter Groups Under the VSP Configuration

The **diameter-group** object defines one or more Diameter servers for Diameter clients. You will need to configure the Diameter group on each OS-E system across one or more clusters where route server call rates are necessary.

CLI Session

```
NNOS-E> config vsp
config vsp> config diameter-group 1
```

```
Creating 'diameter-group 1'  
config diameter-group 1> set admin enabled  
config diameter-group 1> set application routing  
config diameter-group 1> set origin-host domain-name  
config diameter-group 1> set origin-realm domain-name  
config diameter-group 1> config server 1.1.1.1  
config server 1> set admin enabled  
config server 1> set port 3686
```

Configuration Notes

The **application** property specifies the target application for the servers in this Diameter group. Choose **routing** for least-cost-routing between clusters.

The **origin-host** property specifies the text that the OS-E writes to the Origin-Host attribute field in any Diameter requests it sends. This should be the DNS name of the OS-E system you are configuring.

The **origin-realm** property specifies the text that the OS-E system writes to the Origin-Realm attribute field in any Diameter requests it sends. This should be the domain name to which a Diameter group of servers belong.

The **server** object identifies and configures the Diameter server(s) that are part of this Diameter group. Enter a host name or IP address to identify each server.

Configuring the Route Server Service

On each OS-E system in a cluster where route server call rates are necessary, you will need to configure the route server service in the session configuration **authorization** object to control how route server call rate lookups are performed.

CLI Session

```
NNOS-E> config vsp default-session-config authorization  
config authorization> set mode diameter  
config authorization> set always-perform-lookup true  
config authorization> set apply-to-methods INVITE
```

Configuration Notes

The **mode** property sets the method to use for authorization data retrieval. Select **diameter** to use the route server engine.

The **always-perform-lookup** property specifies whether the system should do an authorization lookup. If set to **true**, the OS-E retrieves authorization data regardless of other configuration settings.

Configuring Enterprise Servers and Carriers

For the OS-E to forward calls to the proper end points that are returned by the route server, the end points should be configured on each OS-E system. Depending on your network, you can either configure the enterprise server or the carriers to establish the proper end points.

When importing the call rates at the route server import client, you are not required to specify a carrier or endpoint name. In this case, the carrier defaults to "default" and you will need to create a carrier on the OS-E with the name "default."

CLI Session for Enterprise Servers

```
NNOS-E> config vsp enterprise servers
config servers> config sip-gateway 1
config sip-gateway> set carrier 1
config sip-gateway> config server-pool
config server-pool> config server 1
config server 1> set endpoint default
config server 1> set host 1.1.1.1
```

CLI Session for Carriers

```
NNOS-E> config vsp carriers
config carriers> config carrier 1
Creating 'carrier 1'
config carrier 1> set carrier default
```

Configuring Routing Arbitration

Although route arbitration is not part of the route server, any configured arbiters are applied to the routes returned from the route server on the OS-E. You can configure and edit arbiters under the **vsp\dial-plan** configuration object.

Using the Import Client Features

This section covers the import client functions available from the left pane and from the horizontal tab at the top of the page: **Status, Rates, DID Mapping, Update LCR Server, Log, and Users.**

Import Rates

The route server supports multi-stage routing. You can upload multiple route files to the import tool, update them to the route server, and fetch them back to the import tool to view and edit. In releases previous to 3.6M5, the OS-E supported only one active route file being uploaded and sent to the route server.

When updating multiple files to the route server, you must have the correct number of tables under the **master-services > route-server > table-config** object configured. When importing files to the import tool, you must name the files to match their corresponding previously configured tables.

To configure tables:

1. Select the **Services** tab and click on the **master-services > route-server** object.
2. Click **Add table-config**.

The screenshot shows the Oracle Communications Application Session Controller (OS-C) GUI. The top navigation bar includes 'Home', 'Configuration', 'Status', 'Call Logs', 'Event Logs', 'Actions', 'Services', and 'Keys'. The left sidebar shows a tree view under 'Services: all' with 'route-server' selected. The main content area is titled 'Configure master-services/route-server' and contains several configuration fields: 'admin' (set to 'enabled'), 'host-box' (set to 'cluster/box 1'), and 'group' (set to '0'). Below these fields is a table for 'table-config' with columns for 'table-config', 'description', and 'filename'. The table lists three entries: 'table-config Table1' (description: 'for customer 1', filename: 'file 1'), 'table-config Table2' (description: 'for customer 2', filename: 'file 2'), and 'table-config Table3' (description: 'for customer 3', filename: 'did-file3'). There are 'Edit' and 'Delete' links for each row, and an 'Add table-config' link at the bottom.

table-config	description	filename
table-config Table1	for customer 1	file 1
table-config Table2	for customer 2	file 2
table-config Table3	for customer 3	did-file3

3. Enter the name you want to give the table and the corresponding filename. Click **Create**.

NOTE: This filename must match the corresponding file uploaded to the import tool. When you import the route file into the import tool, ensure you give the file the same filename as you have given it in the corresponding route server table.

4. Click **Set** and update and save the configuration.

You can view the configured tables in the route server via the **show route-server-table-config** status action.

```
NNOS-E>show route-server-table-config
```

```
table                filename                description
-----                -
Table1               file1                   for customer1
Table2               file2                   for customer 2
Table3               did-file3               for customer 3
```

Field	Description
table	Name of the table.
filename	Name of the table's associated route file.
description	Description of the table.

The **Rates** tab allows you to import rate tables, view imported LATA and region code tables, and purge all of the rate tables.

When you click on the **Rates Ranges** tab, the following is displayed in the left pane. For users with read-only access, the **Backup**, **Restore**, **Purge rates**, **Purge templates**, and **Purge route files** links are disabled.

```
Manage
  Rates
  LATA
  Region Code

Backup
Restore

Purge rates
Purge Templates

Purge route files
```

Import Rates — Step 1

Use the Import Rates page to import rate tables provided by carriers into the import client. To access this page, select the **Rates** tab and click the **Import** button.

There are four steps associated with this task. Step 1 requires that you locate and specify the name of the rate file, as well as configure other settings.

To select a rate file to import:

1. **Format**—Specify **Other** or **L3**.
2. **File**—Browse to the directory containing the rate file in CSV format, as provided by your carrier.
3. **Do the file(s) contain a header line?**—Specify **Yes** or **No**. Select **Yes** if the file has a header line.
4. **File Delimiter**—Enter the delimiter for the file. The default setting is the comma (.).
5. **Do the file(s) need a conversion?**—Make one of the following choices:
 - **No**—No conversion necessary.
 - **Yes, LATA to NPA/NXX**—One of the columns contains a LATA (Local Access Transport Areas) number that will require conversion to an NPA/NXX number. NPA-NXX codes refer to the combination of area codes (NPAs) and local exchanges (NXXs). The combined code may contain up to 10,000 telephone numbers (the last 4 digits) that are usually located within a specific geographic region associated with the Central Office of that code.

Example: 212-555 is the NPA-NXX for the phone number 212-555-1212

NPA-NXXs provide a mechanism for geocoding direct mail lists, particularly for the telecommunications industry.



Note: If you select the LATA to NPA/NXX option, you will need to run the Import LATA function from the left pane before proceeding.

- **Yes, Region Code to Dial Code**—One of the columns contains a region code that will require conversion to a dial code number.
6. **Generate 'From URL' field**—Make one of the following choices:

- **Don't Generate**—The fromURL field does not need to be generated.
 - **For Contiguous U.S.**—For every row in the file the fromURL field will be filled in with the NPA for the contiguous U.S...
 - **For Contiguous U.S. + Hawaii**—For every row in the file the fromURL field will be filled in with the NPA for the contiguous U.S. + Hawaii.
7. **Skip empty rate rows**—Ignores rows with empty rate values.
 8. **Use a template for step 2 parameters**—Specify the name of a previously-created CSV column mappings template from the route server Import Rates—Step 2 page. Leave this field blank if no templates exist.

The screenshot shows the 'Import Rates - Step 1' configuration page. The page has a navigation bar with tabs for 'Status', 'Rates', 'DID Ranges', 'Route Server', and 'Audit Log'. On the left, there is a 'Manage' section with links for 'Rates', 'LATA', 'Region Code', 'Backup', 'Restore', 'Purge Rates', and 'Purge Templates'. The main content area contains the following settings:

- Format:** Other (dropdown)
- File:** [Text input field] [Browse... button]
- Do the file(s) contain a header line?** Yes (dropdown)
- File delimiter:** . (text input)
- Do the file(s) need a conversion?** No (dropdown)
- Generate 'From URL' field:** Don't Generate (dropdown)
- Skip empty rate rows?** No (dropdown)
- Use a template for step 2 parameters:** [Dropdown menu]

A 'Next' button is located at the bottom right of the configuration area.

Click **Next** to proceed to Step 2.

Import Rates — Step 2

Step 2 requires you to enter the column mappings. The Oracle values are listed on the left and required parameters are marked with a '*'. For each parameter you can either choose a value from the CSV file or you can enter a fixed value that applies to all records.

acme packet Logout Help
About LCR Import

Status Rates DID Ranges Route Server Audit Log

Import Rates - Step 2 - Column Assignment

CSV column selected and fixed value specified. Select either a value from the CSV file field or specify a fixed value.

Value	CSV file Field	Fixed Value
*to-URL-match	<input type="text"/>	<input type="text"/>
*rate	<input type="text"/>	<input type="text"/> (cents per minute)
effective-start-date	<input type="text"/>	<input type="text"/> yyyy-MM-dd HH:mm:ss
effective-end-date	<input type="text"/>	<input type="text"/> yyyy-MM-dd HH:mm:ss
carrier	<input type="text"/>	<input type="text"/> (default: 'default')
endpoint	<input type="text"/>	<input type="text"/> (default: 'default')
from-URL-match	<input type="text"/>	<input type="text"/>
emergency	<input type="text"/>	<input type="text"/>
priority	<input type="text"/>	<input type="text"/> 1 - 100 (High - Low)
minimum-duration	<input type="text"/>	<input type="text"/> (seconds)
billing-increment	<input type="text"/>	<input type="text"/> (seconds)
from-URL-alteration	<input type="text"/>	<input type="text"/> sip.username@domain
to-URL-alteration	<input type="text"/>	<input type="text"/> sip.username@domain
request-alteration	<input type="text"/>	<input type="text"/> sip.username@domain
passert-alteration	<input type="text"/>	<input type="text"/> sip.username@domain
session-config-name	<input type="text"/>	<input type="text"/>
days	<input type="text"/>	<input type="text"/> sun mon tue wed
start-time	<input type="text"/>	<input type="text"/> HH:mm:ss
end-time	<input type="text"/>	<input type="text"/> HH:mm:ss
Timezone for start/end times	enter GMT-5:00 or select from GMT-minus-5	
min-digits	<input type="text"/>	<input type="text"/> (0 - no minimum specified)
max-digits	<input type="text"/>	<input type="text"/> (0 - no maximum specified)
variables	<input type="text"/>	<input type="text"/> Name Value Remove

Save as template

To select column mappings:

1. ***to-URL-match**—The To: URL field. If a conversion is required, then use the same field as the conversion.
2. ***rate**—The call rate for this record.
3. **effective-start-date**—The start date for this record.
4. **effective-end-date**—The end date for this record.
5. **carrier**—The carrier name.
6. **endpoint**—The endpoint name. This name will need to be in your configuration.
7. **from-URL-match**—The From: URL field.

8. **emergency**—Specify **True** or **False**.
9. **priority** — The priority for this record.
10. **minimum-duration**—The minimum duration for this record.
11. **billing-increment**—Billing increment.
12. **from-URL-alteration**—Alters the from-uri-specification.
13. **to-URL-alteration**—Alters the to-uri-specification.
14. **request-alteration**—The Request URI header.
15. **passert-alteration**—The number in the P-Asserted-Identity field.
16. **session-config-name**—Specifies a session-config-pool entry from the OS-E configuration.
17. **days**—The selected day(s) of the week when the rate is valid.
18. **start-time**—The daily time at which the rate import starts.
19. **end-time**—The daily time at which the rate import ends.
20. **Timezone for start/end times**—Greenwich Mean Time (GMT) plus or minus the number of hours to the current time zone.
21. **min-digits**—The minimum digits for this record. A value of **0** means there is no minimum.
22. **max-digits**—The maximum digits for this record. A value of **0** means there is no maximum.
23. **variables**—Specify the variables field of the CSV file. Variable values supersede the CSV file field value if both are specified.
24. **Save as template**—Saves the current column mappings to a named file. Once created, you can specify the named template on the Import Rates—Step 1 page.
25. Click **Next** to proceed to Step 3.

Import Rates — Step 3

Step 3 confirms that rates have been converted, reports any conversion errors, shows the number of rate records to be imported, and the number of failures (if any).

acme packet Logout Help
About LCR Import

Status Rates DID Ranges Route Server Audit Log

Import Rates - Step 3 - Confirmation

Successfully read .csv file(s).

of Records: 2 Failures: 0 Skipped: 0

The table below shows a subset of what will be imported. Please verify the column mappings before importing.

to-URL-match	from-URL-match	carrier	endpoint	emergency	priority	rate	effective-start-date	effective-end-date	session-config-name	days	start-time	end-time	min-digits	max-digits
2012080	.*	default	internal	false	100	2012089				all			0	0
2012092	.*	default	internal	false	100	2012094				all			0	0

Cancel Back Import

A sample table displays up to five records, allowing you to verify that the column mappings are correct before importing. Click the **Import** button to import the records into the route server Import client database.

Import Rates — Step 4

Once the rates have been successfully imported, Step 4 updates the route server from the import client. Select **Yes** from the drop-down menu, then **Finish**.

If you need to go back to Step 3 to make changes, select **Change Column Assignments**.

acme packet Logout Help
About LCR Import

Status Rates DID Ranges Route Server Audit Log

Import Rates - Step 4 - Update LCR Server

Successfully imported. You will need to update the route server for these rates to take effect.

Inserted: 0 Updated: 2 Failures: 0

Update the route server? No

Import LATA

The **Import LATA** page allows you to import a LATA to an NPA/NXX table. Some rate tables only contain the LATA field. The import rates view will use this information to convert a LATA to an NPA/NXX. The CSV file should contain at least three columns. You must enter the column indexes for the LATA, NPA, and NXX columns.

To import LATA data:

1. Select the **Rates** tab.
2. From the left pane of the Rates page, select **LATA** to display the Import LATA to NPA.NXX Data.
3. Click **Import**.

4. **Purge Existing Records?**—Specify **Yes** or **No**. The default is **No**.
5. **Format**—Specify **LERG10 Data File** or **CSV File**. The LERG10.dat file is available from the following Web URL:
http://www.telcordia.com/products_services/trainfo/catalog_details.html
6. **File**—Browse to the directory containing the rate file in LERG10 or CSV format, as provided by your carrier.

Click the **Import** button to import the LATA records into the import client database. This operation may take a few minutes to complete. When the page refreshes, a message tells you that the LATA data has been successfully imported to the route server import client.

Import Region Code

This view allows a region code to dial code table to be imported. Some rates tables only contain the region code field. The import rates view uses this information to convert a region code to a dial code. The CSV file must contain at least two columns, region code and dial code.

To import Region Code data:

1. Select the **Rates** tab.
2. From the left pane of the Rates page, select **Region Code**.
3. Click **Import** to display the Import Region Code to Dial Code Data.

The screenshot shows the 'Import Region Code to Dial Code Data' form. The form has a left sidebar with navigation links: Manage, Rates, LATA, Region Code, Backup, Restore, Purge Rates, and Purge Templates. The main area contains the following fields:

- Purge Existing Records?: No (dropdown)
- File: [text input] Browse...
- Does the file contain a header line?: Yes (dropdown)
- File delimiter: [text input] ,
- Region Code Column #: [text input]
- Dial Code Column #: [text input]
- Import (button)

4. **Purge Existing Records?**—Specify **Yes** or **No**. The default is **No**.
5. **File**—Browse to the directory containing the region code file in CSV format, as provided by your carrier.
6. **Does this file contain a header line?**—Specify **Yes** or **No**. Select **Yes** if the file has a header line.
7. **File Delimiter**—Enter the delimiter for the file. The default setting is the comma (,).
8. **Region Code column #**—Enter the column number in the file containing the region code.
9. **Dial Code Column #**—Enter the column number in the file containing the dial code.

Click the **Import** button to import the region codes into the import client database. This operation may take a few minutes to complete. When the page refreshes, a message tells you the number of region codes that have been inserted and updated, as well as the number of failures that occurred while importing this data.

Backing Up Rates

The route server import tool backup functionality allows you to save imported rates to either a compressed XML or CSV file.

To backup rates:

1. Select the **Rates** tab.
2. From the left pane of the Rates page, click **Backup**.

3. **Backup file format**—Select the type of file you want to back up, either **XML** or **CSV**. The default setting is **XML**.
4. **File name**—Enter a name for this file. By default it is called `carrier_routing_YYYY-MM-DD_HH-MM` where `YYYY-MM-DD_HH-MM` is the date and time of the file download.
5. Click **Backup** to finish.

When the backup is complete, the archive is stored in the OS-E and a message similar to the following example is displayed:

```
Successfully saved rates to file:
  carrier_routing_2016_01_08_13-17.xml.gz
```

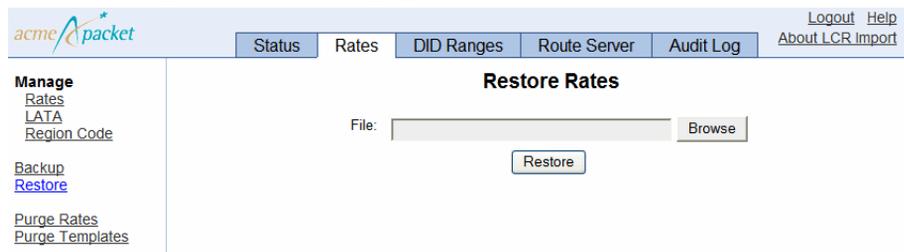
Restoring Rates

The restore function allows you to take advantage of existing call rates that currently exist on the OS-E route server. If you currently have a route server import client and route server, you can fetch the call rates from the route server to restore the route files on a new or existing route server import client.

From the route server, locate the carrier routing file from the directory **/cxc_common/lcrimport/lcrimportdata/backups/**. The file that restores a route server import client is called `carrier_routing_<date_timestamp>.xml.gz`.

To restore a route server import tool:

1. From the OS-E Management System, use the **Tools** tab to move the file from the route server to the route server import client.
2. Select the **Rates** tab.
3. From the left pane of the Rates page, click **Restore**.



4. Click **Browse** and select the `carrier_routing_<date_timestamp>.xml.gz` file.
5. Click **Restore**. When the page refreshes, a message tells you the number of rates that have been inserted and updated, as well as the number of failures that occurred while restoring rates to the route server client.

Purging From the Route Server

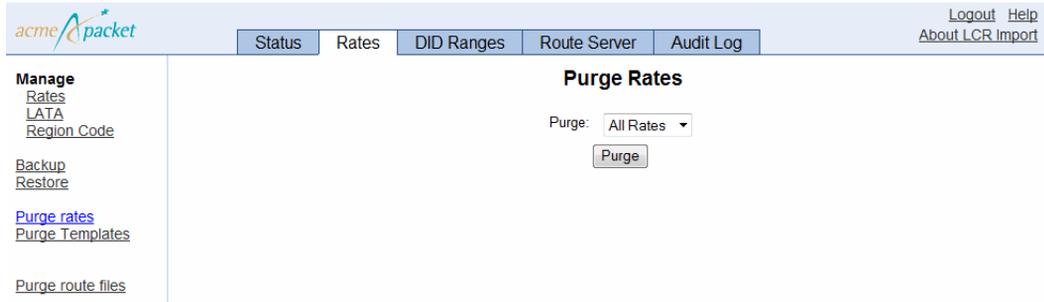
Within both the Rates and DID Ranges pages you have the ability to purge rates, templates, and route files.

Purging Rates

From the Rates page, you can purge rates, templates, or route files.

To purge rates:

1. Select the **Rates** tab.
2. Click **Purge rates** from the left pane.



3. **Purge**—Select either **All Rates** to purge all saved rates or **Limited to** to purge a rate range. The default setting is **All Rates**.

When you select **Limited to**, several parameters appear which allow you to specify the criteria on which to purge.

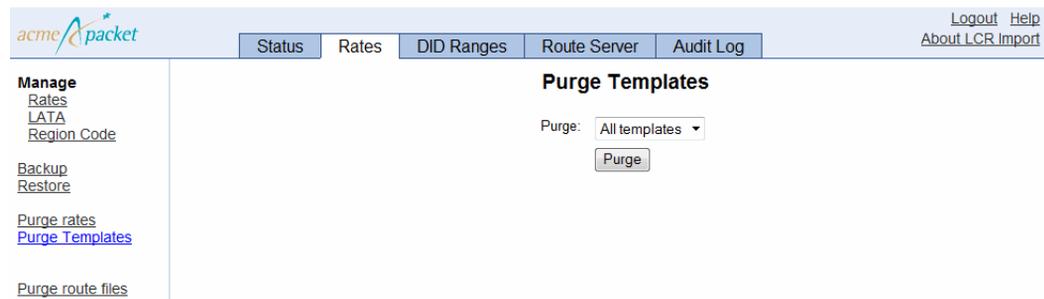
4. Click **Purge**.

A pop-up appears explaining that purging route files irreversibly deletes existing rates and entries from the database and all local route files.

5. Click **OK**.

To purge templates:

1. Select the **Rates** tab.
2. Click **Purge Templates** from the left pane.



3. **Purge**—Select either **All templates** to purge all saved templates or **Limited to** to purge a specific template. The default setting is **All templates**.

When you select **Limited to**, the **Template name** parameter appears. From the drop-down list, select which template to purge.

4. Click **Purge**.

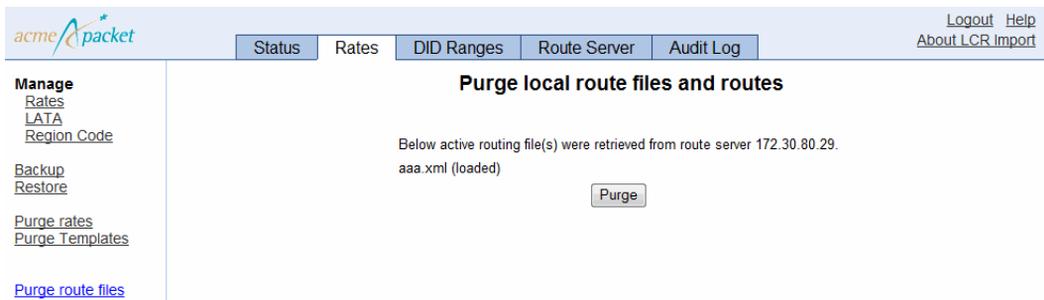
A pop-up appears explaining that purging templates is irreversible.

5. Click **OK**.

To purge route files:

1. Select the **Rates** tab.

2. Click **Purge route files** from the left pane.



3. Select the file you want to purge if there are multiple route files saved to the route server import tool.

4. Click **Purge**.

A pop-up appears explaining that purging route files irreversibly deletes existing rates and entries from the database and all local route files.

5. Click **OK**.

Purging DIDs

From the DID Ranges page, you can purge DID ranges, templates, or route files.

To purge DID Ranges:

1. Select the **DID Ranges** tab.

- Click **Purge DID ranges** from the left pane.

The screenshot shows the 'acme packet' web interface. At the top, there are navigation tabs: 'Status', 'Rates', 'DID Ranges', 'Route Server', and 'Audit Log'. In the top right corner, there are links for 'Logout', 'Help', and 'About LCR Import'. On the left side, there is a sidebar with the following links: 'Manage DID ranges', 'Edit', 'Backup', 'Restore', 'Purge DID ranges', 'Purge templates', and 'Purge route files'. The main content area is titled 'Purge DIDs' and contains a 'Purge:' dropdown menu currently set to 'All DIDs' and a 'Purge' button below it.

- Purge**—Select either **All DIDs** to purge all saved DID ranges or **Limited to** to purge a DID rate range. The default setting is **All DIDs**.

When you select **Limited to**, several parameters appear which allow you to specify the criteria on which to purge.

- Click **Purge**.

A pop-up appears explaining that purging route files irreversibly deletes existing rates and entries from the database and all local route files.

- Click **OK**.

To purge templates:

- Select the **DID Ranges** tab.
- Click **Purge Templates** from the left pane.

The screenshot shows the 'acme packet' web interface. At the top, there are navigation tabs: 'Status', 'Rates', 'DID Ranges', 'Route Server', and 'Audit Log'. In the top right corner, there are links for 'Logout', 'Help', and 'About LCR Import'. On the left side, there is a sidebar with the following links: 'Manage DID ranges', 'Edit', 'Backup', 'Restore', 'Purge DID ranges', 'Purge templates', and 'Purge route files'. The main content area is titled 'Purge DID Templates' and contains a 'Purge:' dropdown menu currently set to 'All templates' and a 'Purge' button below it.

- Purge**—Select either **All templates** to purge all saved templates or **Limited to** to purge a specific template. The default setting is **All templates**.

When you select **Limited to**, the **Template name** parameter appears. From the drop-down list, select which template to purge.

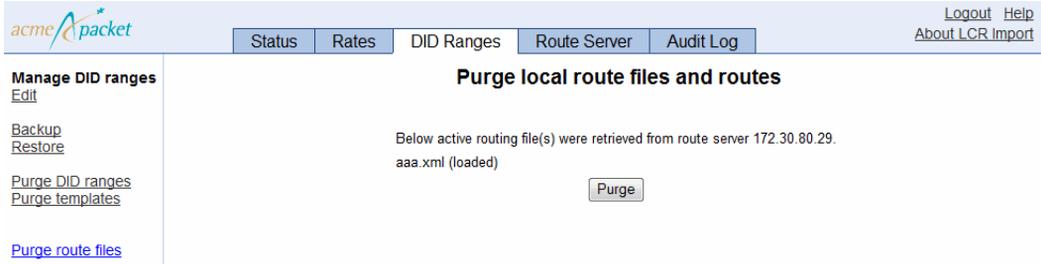
- Click **Purge**.

A pop-up appears explaining that purging templates is irreversible.

5. Click **OK**.

To purge route files:

1. Select the **DID Ranges** tab.
2. Click **Purge route files** from the left pane.



3. Select the file you want to purge if there are multiple route files saved to the route server import tool.
4. Click **Purge**.

A pop-up appears explaining that purging route files irreversibly deletes existing rates and entries from the database and all local route files.

5. Click **OK**.

Configuring DID Mapping

Direct inward dial (DID) mapping is supported on the route server. The DID Ranges tab allows you to import and configure routes to be entered to the route server import client.

The route server import tool validates all DID ranges during import, add, edit, split, and replace functions. The following are the validation rules.

- Remove any trailing minus “-” and plus “+” signs in DID ranges.
- Required fields are **did-range-start**, **did-range-end**, and **carrier**. These values cannot be null or left empty.
- The DID range start value cannot be greater than the DID range end value.

- Any alpha prefix for DID range start and end must be the same.
- The only situation where you can import or add DID ranges which match existing DID routes, is a range that contains only one number, meaning the did-start-range equals the did-end-range.

When you click on the **DID Ranges** tab, the following is displayed in the left pane. For users with read-only access, the **Backup**, **Restore**, **Purge rates**, **Purge templates**, and **Purge route files** links are disabled.

Manage DID ranges

[Edit](#)

[Backup](#)

[Restore](#)

[Purge DID ranges](#)

[Purge templates](#)

[Purge route files](#)

When you click on the **DID Ranges** tab, the default display in the right panel is the **Edit** display for viewing and editing DID ranges. Once you select a DID entry, the **Split**, **Edit**, and **Delete** buttons also become available.

Users who have read-only access may read the existing records, however, they will not be able to perform any modifications and the **Import**, **Add**, **Split**, **Edit**, **Replace**, and **Delete** buttons are inactive.

Importing DIDs

The **DID Ranges** tab allows you to import DID ranges in a CSV file, backup DID configurations, restore a previous backup into the local database, view, edit, delete, create, and split DID mappings, and purge DID mappings and templates.

Importing DIDs — Step 1

There are five steps associated with importing DIDs. Step 1 requires that you locate and specify the name of the CSV file, as well as configure other settings.

To specify the DID file to import:

1. Select the **DID Ranges** tab.
2. Click **Import**.

3. **File**—Browse to the directory containing the CSV DID file you are importing, provided by your carrier.
4. **Does the file contain a header line?**—Specify **yes** or **no**. **Yes** means the DID file has a header line. **No** means there is no header line and in Step 2 you assign the columns by number.
5. **File delimiter**—Enter the delimiter for the file. The default setting is the comma (,).
6. **Use a template for step 2 and 3 parameters**—Specify a previously-created CSV columns mapping template if you have one configured and want to use it. Leave this field blank if you are not using a template.

7. Click **Next** to proceed to Step 2.

Importing DIDs — Step 2

In Step 2 map the columns in your CSV file to the route server import properties. If a header is present in the CSV file, the column headings can be used in the mapping. Otherwise, column numbers starting with 0 are used for mapping.

acme packet Logout Documentation
About LCR Import

Status Rates DID Ranges Route Server Audit Log

Import DIDs - Step 2 Column Assignment

Select a value from the CSV file field or specify a fixed value. Note: The mandatory fields are did-range-start and did-range-end properties.

Value	CSV file Field	Fixed Value
* did-range-start	(Select Field)	
* did-range-end	(Select Field)	
description	(Select Field)	
from-URL-match	(Select Field)	(default: ".")
* carrier	(Select Field)	(default: 'default')
endpoint	(Select Field)	(default: 'default')
priority	(Select Field) <input checked="" type="checkbox"/> Ignore	65535 (default: '100', (range: 0 - 65534, High - Low), (ignore: 65535))
from-URL-alteration	(Select Field)	(example: sip.username@domain)
to-URL-alteration	(Select Field)	(example: sip.username@domain)
request-alteration	(Select Field)	(example: sip.username@domain)
passert-alteration	(Select Field)	(example: sip.username@domain)
session-config-name	(Select Field)	
variables	(Select Field)	Add variable

Use only did-range-start and did-range-end to make DID unique? Yes

Save as template

Cancel Back Next

1. **did-range-start**—Select the column from the CSV file that represents the start value of the DID range. This is a required field.
2. **did-range-end**—Select the column from the CSV file that represents the end value of the DID range. This is a required field.
3. **description**—Enter a description of this DID range.
4. **from-URL-match**—Select the column from the CSV file that represents the From: URL field.
5. **carrier**—Select the column from the CSV file that represents the carrier name. This is a required field.

6. **endpoint**—Select the column from the CSV file that represents the endpoint name. This name needs to be in your configuration already.
7. **priority**—Specify the priority of this DID range. If this value is between 0-65534, this value takes precedence over the sip-gateway server preference for server arbitration. If this value is 65535, the sip-gateway server preference takes precedence.

By enabling the **Ignore** checkbox, the import client ignores the configured priority and assigns a default value of 65535.
8. **from-URL-alteration**—Select the column from the CSV file that represents the alterations of the from-uri-specification.
9. **to-URL-alteration**—Select the column from the CSV file that represents alterations of the to-url-specification.
10. **request-alteration**—Select the column from the CSV file that represents the Request URI header.
11. **passert-alteration**—Select the column from the CSV file that represents the number in the P-Asserted-Identity field.
12. **session-config-name**—Select the column from the CSV file that represents a **session-config-pool** entry from the OS-E configuration.
13. **variables**—Specify the variables field of the CSV file. Variable values supersede the CSV file field value if both are specified.
14. **Use only did-range-start and did-range-end to make DID unique?**—Select **Yes** if you want to only use the **did-range-start** and **did-range-end** values for matching.
15. **Save as template**—Saves the current column mappings to a name you specify. Once created, you can specify the named template in the Importing DID Ranges — Step 1.
16. Click **Next** to proceed to Step 3.

Importing DIDs — Step 3

In Step 3 the route server import reads the CSV file, reports any issues, and displays sample records for verifying that the CSV column to DID route record column assignment is correct. On this page you can also make any necessary translations using regular expressions.

Click **Next** to proceed to Step 4.

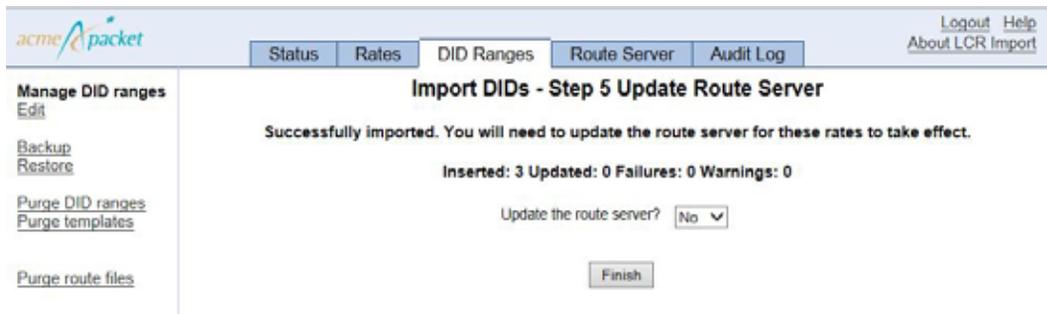
Importing DIDs — Step 4

In Step 4, the route server import tool displays a preview of what is going to be imported after you have made any translations.

Click **Import**. The route server import stores DID records constructed from the CSV file into the database

Importing DIDs — Step 5

In Step 5, you have the option to **Finish** the import, or to continue importing files.



The screenshot shows the Oracle Communications Application Session Controller interface. The top navigation bar includes tabs for Status, Rates, DID Ranges, Route Server, and Audit Log. The main content area is titled "Import DIDs - Step 5 Update Route Server" and displays a success message: "Successfully imported. You will need to update the route server for these rates to take effect." Below the message, it shows statistics: "Inserted: 3 Updated: 0 Failures: 0 Warnings: 0". There is a dropdown menu for "Update the route server?" with "No" selected. A "Finish" button is visible at the bottom.

Once all imports are complete, you can then update the route server by going to the **Route Server** tab. If you choose not to update the route server, clicking **Finish** returns you to the **DID Ranges** Edit/View display.

Searching DID Ranges

To narrow the view of what is displayed in the DID Ranges tab, click the **Search** button.

You can specify “*” as a wildcard in any position in the search string. This wildcard can match 0 or more characters. Below is the resulting filtered view from the above search.



Click the **View All** button to restore the DID Ranges tab to displaying all of the DID ranges present in the route server import database.

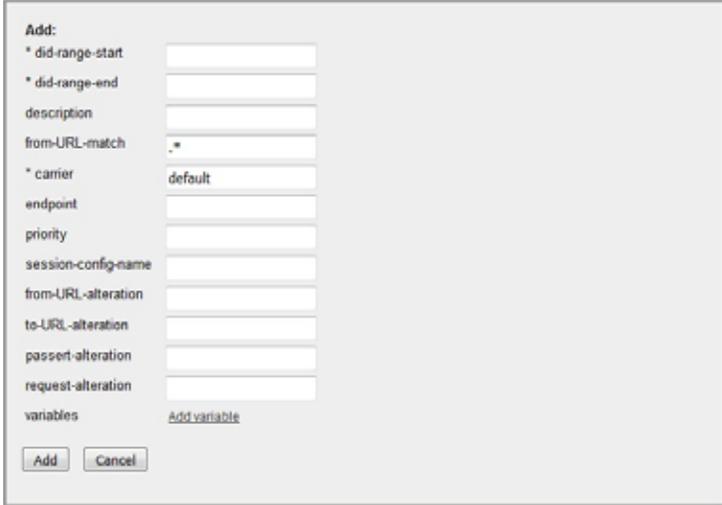
Adding DID Ranges

You can manually add DID ranges to a route file.

To add a DID range:

1. Select the **DID Ranges** tab.
2. Load the route file to which you are adding a DID range.

3. Click the **Add** button.



Add:	
* did-range-start	<input type="text"/>
* did-range-end	<input type="text"/>
description	<input type="text"/>
from-URL-match	<input type="text" value="*"/>
* carrier	<input type="text" value="default"/>
endpoint	<input type="text"/>
priority	<input type="text"/>
session-config-name	<input type="text"/>
from-URL-alteration	<input type="text"/>
to-URL-alteration	<input type="text"/>
passert-alteration	<input type="text"/>
request-alteration	<input type="text"/>
variables	Add variable
<input type="button" value="Add"/> <input type="button" value="Cancel"/>	

The **carrier** and **from-URL-match** properties are populated with default values. The only required fields to add a DID range are **did-range-start**, **did-range-end**, and **carrier**.

If you attempt to add a duplicate or overlapping DID range with a record already stored in the database, you receive an error message from the route server import and the range is not added to the database.

Splitting DID Ranges

The route server allows you to split a single existing DID range into two or more individual ranges.

To split a DID range:

1. Select the **DID Ranges** tab.
2. Load the route file on which you are splitting a DID range.
3. Select the DID route record you want to divide and click the **Split** button.

Only one split row can be added at a time, however, multiple split ranges within the same DID range are supported.

If you enter all **DID Range Start** values, the route server import tool calculates the **DID Range End** automatically, or vice versa. You cannot, however, enter consecutive blank **DID Range Start** and **DID Range End** values.

If the selected DID range has an alphabetical prefix, the new ranges contain the same prefix.

Click **Add** to complete the splitting process.

Editing DID Ranges

You can access a DID range and manually make edits.

To edit a DID range

1. Select the **DID Ranges** tab.
2. Load the route file on which you are editing
3. Select the DID route record and click **Edit**.

After you've made the changes you want to make, click **Save** to complete the editing process.

Replacing DID Ranges

The DID replacing functionality gives you the ability to select DID ranges using a search criteria, and then replace values in that selection with new values.

To replace DID Ranges:

1. Select the **DID Ranges** tab.
2. Load the route file on which you are replacing values.

- Specify the search criteria along with replacement values for each DID range in the selection.

Search for:		Replace:	
did-range-start	<input type="text"/>	did-range-start	<input type="text"/>
did-range-end	<input type="text"/>	did-range-end	<input type="text"/>
description	<input type="text"/>	description	<input type="text"/>
from-URL-match	<input type="text"/>	from-URL-match	<input type="text"/>
carrier	<input type="text"/>	carrier	<input type="text"/>
endpoint	<input type="text"/>	endpoint	<input type="text"/>
		session-config-name	<input type="text"/>
		variables	Add variable
<input type="button" value="Search"/> <input type="button" value="Cancel"/>			

- Click **Search**.

The route server import displays the number of records found and asks for confirmation.

<p>Your search has returned 13 entries Press 'Replace' button if you want to proceed with replace.</p> <p> <input type="button" value="Replace"/> <input type="button" value="Cancel"/> </p>

- Click **Replace** and the route server import replaces values with the new values you specified.

Deleting a Range Route Record

The DID delete functionality gives you the ability to manually delete a DID route record from a route file.

To delete a DID route record:

- Select the **DID Ranges** tab.
- Load the route file from which you are deleting a record.
- Select the record in the display and click Delete.

The following confirmation appears.

Are you sure you want to delete the selected row?

Click **Yes** to complete the deletion process.

Backing Up DID Ranges

You can backup DID ranges using either an XML or CSV file. By default, the route server import tool names the XML file with the date and time. If you want to name your file differently you can change this.

You can backup either all of the DID ranges or only a subset of DID ranges based on specified criteria to a CSV file.

To backup all DID ranges:

1. Select the **DID Ranges** tab.
2. From the left pane of the DID Ranges page, click **Backup**.

The screenshot shows the Oracle Communications Application Session Controller interface. At the top, there are navigation tabs: Status, Rates, DID Ranges (selected), Route Server, and Audit Log. On the right, there are links for Logout, Help, and About LCR Import. The main content area is titled 'Backup DIDs' and contains the following text: 'The backup action will export data to either an XML or CSV file. Only the XML format can be used in the restore action. When exporting as CSV you may select a subset of the records.' Below this text, there is a 'Backup file format:' dropdown menu set to 'XML', a 'File name:' text input field containing 'did_2016-01-12_14-55', and a 'Backup' button. On the left side of the interface, there is a sidebar menu with options: Manage DID ranges (with sub-links Edit, Backup, Restore), Purge DID ranges, Purge templates, and Purge route files.

3. Click **Backup** to complete the backup process. When the backup completes the route server import tool displays a confirmation on the screen.

To backup a DID range:

- If both **did-range-start** and **did-range-end** are specified, all DID ranges that match the DID range criteria will be included in the CSV file.

The screenshot shows the 'Backup DIDs' interface in the acme packet system. The top navigation bar includes 'Status', 'Rates', 'DID Ranges', 'Route Server', and 'Audit Log'. The left sidebar has links for 'Manage DID ranges', 'Backup', 'Restore', 'Purge DID ranges', 'Purge templates', and 'Purge route files'. The main content area contains the following form:

Backup DIDs

The backup action will export data to either an XML or CSV file. Only the XML format can be used in the restore action. When exporting as CSV you may select a subset of the records.

Backup file format:

File name:

did-range-start:

did-range-end:

description:

from-URL-match:

carrier:

endpoint:

- Click **Backup** to complete the backup process. When the backup completes the route server import tool displays a confirmation on the screen.

Restoring DID Ranges

Depending on how the file has been saved, there are two ways to restore a file onto the route server import tool.

- Restore an XML file using the **Restore** link in the left pane. Browse to the XML file and click **Restore**. For more information on using the **Restore** link, see Restoring Rates.
- Restore a CSV file by reimporting it via the **Import** button. For more information on using the **Import** button, see Importing DIDs.

Updating the Route Server

Once you have imported DID ranges you must update the route server for your changes to take effect.

To update the route server:

- Select the **Route Server** tab.
- Route Server IP**—Enter the route server's IP address. This must be the IP address that has the web-service configuration enabled on the route server.

3. **Route Server port**—Enter the route server’s port. This must be the port that has the **web-service** configuration enabled on the route server.
4. **Web service protocol**—Select the protocol you want to use to transfer files between the route server and route server import tool. This must match the protocol configured in the route server’s web-service.
5. **Remote Authentication**—Specify the authentication type you want to use to transfer files between the route server and route server import tool. This must match the authentication type configured in the route server’s web-service.
6. **Updating using SCP**—Select whether you want to update files using SCP. When set to Yes, you must specify a user name, password, and SSH port. For more information on supported protocols for file transfer, see [HTTPS Support for Call Rates Downloading](#).
7. **Activate routes on the route server?**—You can copy the routes over to the route server without making those routes active. To do that, uncheck the this checkbox. This allows you to test routes via the CLI route-server-test without them being the active route set.
8. **Use default route filename?**—By default, the route server import tool names the route file with the date and time. When this property is enabled, the filename is regenerated with the current date and time. When it is not enabled, the original date and time are left as the name

The screenshot shows the 'Route Server' configuration page. At the top, there is a navigation bar with the 'acme packet' logo on the left and 'Logout Help' and 'About LCR Import' on the right. Below the logo is a menu with 'Status', 'Rates', 'DID Ranges', 'Route Server', and 'Audit Log'. The 'Route Server' tab is selected. The main content area is titled 'Route Server' and contains the following fields:

- Route Server IP:
- Route Server Port:
- Web service protocol:
- Remote Authentication:
- Update using SCP:
- Activate routes on route server?
- Use default route filename?

At the bottom of the form, there are two buttons: 'Update routes' and 'Get active routes'.

9. Click **Update routes** to complete the update process. The display changes to show a progress bar, a description of the current stage of the route server update operation and a **Cancel** button.

Before the update can complete you must confirm a file name for the routes. The route server import displays a default file name, which is the name of the last file retrieved from the route server. You can either leave the default name or overwrite it to something else.

The screenshot shows the 'Route Server' configuration page in the acme packet interface. The navigation bar includes 'Status', 'Rates', 'DID Ranges', 'Route Server', and 'Audit Log'. The 'Route Server' tab is active. The configuration fields are as follows:

- Route Server IP: 172.30.80.29
- Route Server Port: 8080
- Web service protocol: https
- Remote Authentication: certificate
- Update using SCP: No
- Activate routes on route server?
- Use default route filename?

At the bottom, there are two buttons: 'Update routes' and 'Get active routes'.

Click **Continue** to complete the route server update.

Cancelling a Route Server Update

There are two ways to cancel a route server update in progress. You can click the **Cancel** button on the **Route Server** tab.

The screenshot shows the 'Route Server' configuration page with a progress bar indicating an update in progress. The progress bar is at 1% and is labeled 'Updating Route server 172.44.10.78:8787 Writing route file'. A 'Cancel' button is visible below the progress bar.

Or you can click the **Cancel** button under the **Status** tab.

acme packet [Logout](#) [Help](#)
[About LCR Import](#)

Status Rates DID Ranges Route Server Audit Log

Status Seconds

Number of imports in progress: 0
 Number of updates in progress: 1

Update	Status	Action
Route Server: 172.44.10.78:8787	Writing route file	<input type="button" value="Cancel"/>

Number of get route actions in progress: 0
 There is no restore in progress.
 There is no backup in progress.

Retrieving Routes from the Route Server

Via the route server import tool, you can retrieve the list of active route server files from the route server. The route server copies all of the active route set files, deletes all rates and DID ranges from the database, and reads the rate file into the database. These files are transferred via either HTTP, HTTPS, or SCP (depending on what you have configured).

The route server import tool loads the records to the database accordingly, based on whether the route set retrieved from the route server contains all DID ranges, all rates, or a mixture of DID ranges and rates.

To retrieve active route files from the route server to the import tool:

1. Access and log into the route server import tool.
2. Select the **Route Server** tab and click the **Get active routes** button. You are prompted with a pop-up explaining that retrieving routes deletes existing rates and DID entries in the database.
3. Click **OK** to continue.

A list of the active route files running on the route server appears.

The screenshot shows a web interface for 'acme packet' with a navigation bar containing 'Status', 'Rates', 'DID Ranges', 'Route Server', and 'Audit Log'. The 'Route Server' section is active. Under 'Get Active Routing File(s)', the following information is displayed:

Route Server IP:	172.44.10.78
Route Server Port:	8443
Update using SCP:	NO

Below this, there is a checkbox labeled 'Retrieve all active route files listed below from route server?' which is checked. A note says '(Un-check box to retrieve only the selected file.)'. Underneath, it says 'Selected route file will be loaded for editing:' followed by three radio button options: 'file1' (selected), 'file2', and 'carrier-routing-file4'. At the bottom are 'Continue' and 'Cancel' buttons.

- To retrieve all active route files, keep **Retrieve all active route files listed below from route server?** checked. When you retrieve all of the active route files, the file that is selected is the file that gets loaded into the DID Ranges database by default.

To retrieve only one file, uncheck the checkbox and select the file you want. This file is loaded into the DID Range database.

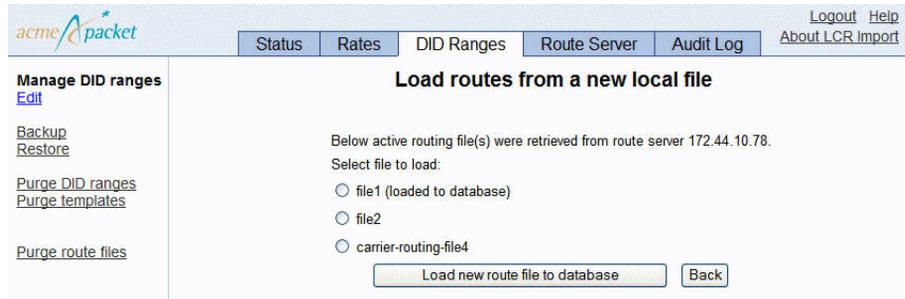
- Click **Continue**. You are prompted with a pop-up explaining that after retrieving the route file, rates and DID entries currently in the database are deleted. Click **OK** to continue.
- Select either the **Rates** or **DID Ranges** tab and you see the selected route file in the database.

The contents of the currently loaded route file is displayed at the top of the DID Ranges and Rates pages. Two buttons on the Rates and DID Ranges pages, **Load** and **Save**, allow you to load and save different route files to the database

Loading and Saving Route Files to the Database

To load a different route file into the database:

1. Click the **Load** button. A list of locally stored route files appears.



2. Select the file to load and click the **Load new route file to database** button.
3. A pop-up appears asking if you want to save the routes currently loaded in the database. Click **OK** to save routes and click **Cancel** if you do not want to save.
4. A pop-up appears explaining that loading a new route file into the database deletes the existing route file from the database. Click **OK** to continue. Upon completion, a status message appears saying the route file was successfully loaded.

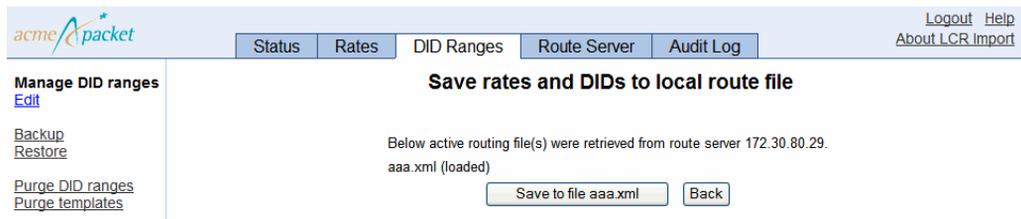
You can save any edits you make to the loaded route file.

To save rate and DID edits to the loaded route file:

1. After you have made your changes, click the **Save** button.

A page appears with information including the IP address of the route server where the route file was retrieved from, as well as the names of the other files retrieved (if applicable).

2. Click the **Save to file <filename>** button to continue. Click **Back** to cancel.



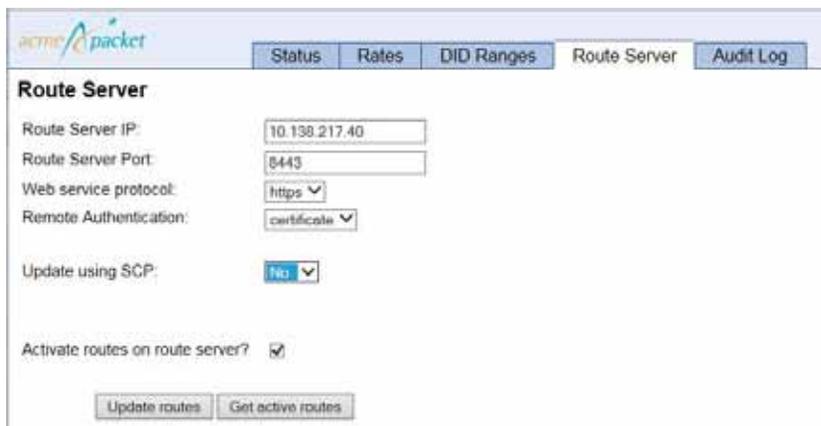
A status message appears confirming you have successfully saved the database entry edits to the route file.

After you finish viewing and/or editing the retrieved route files, you can send them back to the route server.

If only one route file was retrieved and stored locally, you are able to overwrite the original filename before sending it back to the route server. However, if the entire active route file set was retrieved and stored locally, the entire active route file set is sent to the route server for update and in this case, you must keep all of the original filenames.

To send updated route files to the route server:

1. Select the **Route Server** tab. Keep **Activate routes on route server?** checked if you want the routes to be active when you update them. Uncheck this box if you are updating the routes to the route server for testing purposes.
2. Click the **Update routes** button.



The screenshot displays the 'Route Server' configuration interface. At the top, there are navigation tabs: 'Status', 'Rates', 'DID Ranges', 'Route Server' (selected), and 'Audit Log'. Below the tabs, the 'Route Server' section contains the following fields and controls:

- Route Server IP:
- Route Server Port:
- Web service protocol:
- Remote Authentication:
- Update using SCP:
- Activate routes on route server?

At the bottom of the configuration area, there are two buttons: 'Update routes' and 'Get active routes'.

A list of locally stored route files appears.

3. Click **Continue**. A pop-up appears listing the files being sent. Click **OK** to continue. When the files have been updated a message appears explaining that the files were successfully copied to the route server.

Viewing the Status of Route Server Import Operations

The route server import tool status page displays import and route update operations in progress, **Get Route** operations in progress, and current **route-server-controlled-status** status report from the route server.

The screenshot shows the 'acme packet' web interface. At the top, there are navigation tabs: 'Status', 'Rates', 'DID Ranges', 'Route Server', and 'Audit Log'. The 'Status' tab is selected. In the top right corner, there are links for 'Logout', 'Help', and 'About LCR Import'. Below the navigation, there is a 'Status' section with a 'Seconds' input field and a 'Refresh' button. The main content area displays the following information:

- Successfully updated route files [aaa.xml] to server 172.30.80.29:8443 on 2012-06-04 12:13:04.364
- Routes from 172.30.80.29 currently retrieved [aaa.xml (loaded)]
- Number of imports in progress: 0
- Number of local routes updates in progress: 0
- Number of updates in progress: 0
- Number of get route actions in progress: 0
- There is no restore in progress.
- There is no backup in progress.
- Route server status for 172.30.80.29:8443 (route-server-controlled-status)

The route server status is shown in a box with the following details:

```

box 1
master true
start 12:13:04 2012-06-04
end 12:13:04 2012-06-04
action update /cxc_common/aaa.xml "" 20
state inactive
Fetch 12750
routes 23
load-set aaa.xml
activated-at 12:13:04 2012-06-04
box-state Ready
result Success
    
```

Testing DID Ranges and Prefix Changes

You can test imported DID ranges and prefix changes you have made in the route server import tool before you activate them in a live environment. A new action has been created, **route-server-test**, that allows you to test routes, CDRs, and queries, and analyze, compare, and validate results of the routes.

NNOS-E>**route-server-test** ?

Route server test action

```

syntax: route-server-test config [file] [test-vector-file]
       route-server-test cdr file [test-vector-file]
       route-server-test lookup test-vector-file [test-results-file]
       [table]
       route-server-test analyze test-results-file
       [analysis-results-file]
    
```

```

route-server-test compare test-results-file1
test-results-file2 [diff-results-file]
route-server-test validate test-results-file
[validation-results-file] [table]

```

The **route-server-test config** [*routes.xml*] [*test.xml*] action generates a series of test vectors derived from the routes.xml file and outputs them to a specified test.xml file. If you do not specify a test.xml file, the OS-E writes the resulting output to the screen. The test.xml file has the following format.

```

<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="mgmt_data.xsd">
<RouteServerTestSuite suite="1">
  <description>DID 1000-1002</description>
  <tests>
    <RouteServerTestCase>
      <query>1000-1002</query>
      <from/>
      <time></time>
    </RouteServerTestCase>
  </tests>
</RouteServerTestSuite>
</config>

```

The **route-server-test cdr** [*cdr.csv*] [*text.xml*] action generates a series of test vectors derived from accounting records in the CSV format and outputs them to a specified test.xml file. If you do not specify a test.xml file, the OS-E writes the resulting output to the screen. The test.xml file has the following format.

```

<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="mgmt_data.xsd">
<RouteServerTestSuite suite="1">
  <description>DID 1000-1002</description>
  <tests>
    <RouteServerTestCase>
      <query>1000-1002</query>
      <from/>
      <time></time>
    </RouteServerTestCase>
  </tests>
</RouteServerTestSuite>
</config>

```

The CSV file has the following format.

```
"SessionID", "Recorded", "CallID", "To", "From", "Method", "IncomingRequestURI", "PreviousHopIp", "PreviousHopVia", "OutgoingRequestURI", "NextHopIp", "NextHopDn", "Header", "Origin", "SetupTime", "ConnectTime", "DisconnectTime", "DisconnectCause", "Duration", "scpName", "CallID2", "OrigGW", "TermGW", "PacketsReceivedOnSrcLeg", "PacketsLostOnSrcLeg", "PacketsDiscardedOnSrcLeg", "PdvOnSrcLeg", "MaxJitterOnSrcLeg", "CodecOnSrcLeg", "MimeTypeOnSrcLeg", "LatencyOnSrcLeg", "MaxLatencyOnSrcLeg", "RFactorOnSrcLeg", "PacketsReceivedOnDestLeg", "PacketsLostOnDestLeg", "PacketsDiscardedOnDestLeg", "PdvOnDestLeg", "MaxJitterOnDestLeg", "CodecOnDestLeg", "MimeTypeOnDestLeg", "LatencyOnDestLeg", "MaxLatencyOnDestLeg", "RFactorOnDestLeg", "Rx1000FactorOnDestLeg", "Rx1000FactorOnSrcLeg", "MOSFmtOnDestLeg", "MOSFmtOnSrcLeg", "callType", "disconnectErrorType", "ani", "callSourceRegid", "callDestRegid", "newAni", "cdrType", "huntingAttempts", "callPDD", "callSourceRealmName", "callDestRealmName", "callDestCRName", "in_peer_dst", "in_anchor_src", "in_anchor_dst", "in_peer_src", "out_peer_dst", "out_anchor_src", "out_anchor_dst", "out_peer_src", "calledPartyAfterSrcCallingPlan", "lastStatusMessage", "LastMediaPktTimestampOnDestLeg", "LastMediaPktTimestampOnSrcLeg", "SetupTimeInt", "IncomingURIStripped", "dnis", "newDnis", "customData", "CreationTimestamp"
```

The **route-server-test lookup** [*test.xml*] [*result.xml*] [*table*] action uses the test vectors generated from the **route-server-test config** and **route-server-test cdr** actions and queries the route server. The results of the queries are outputted to a specified results.xml file. If you do not specify a result.xml file, the OS-E writes the resulting output to the screen. The result.xml file has the following format.

```
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="mgmt_data.xsd">
<RouteServerTestSuiteResults suite="1">
  <description>DID 1000-1002</description>
  <results>
    <RouteServerTestResult>
      <query>1000</query>
      <from/>
      <time></time>
      <routes>
        <RouteServerTestRouteResult>
          <match>route-plan:2</match>
          <carrier>default</carrier>
          <endpoint>example.net</endpoint>
        </RouteServerTestRouteResult>
      </routes>
    </RouteServerTestResult>
  </results>
  <results>
    <RouteServerTestResult>
      <query>1001</query>
      <from/>
      <time></time>
```

```

    <routes>
      <RouteServerTestRouteResult>
        <match>route-plan:2</match>
        <carrier>default</carrier>
        <endpoint>example.net</endpoint>
      </RouteServerTestRouteResult>
    </routes>
  </RouteServerTestResult>
</results>
<results>
  <RouteServerTestResult>
    <query>1002</query>
    <from/>
    <time></time>
    <routes>
      <RouteServerTestRouteResult>
        <match>route-plan:2</match>
        <carrier>default</carrier>
        <endpoint>example.net</endpoint>
      </RouteServerTestRouteResult>
    </routes>
  </RouteServerTestResult>
</results>
</RouteServerTestSuiteResults>
</config>

```

You can also execute this action with the optional *table* parameter. This allows you to execute the lookup in a different routing table other than the currently active one.

The **route-server-test analyze** [*results.xml*] [*analysis.xml*] action analyzes the results file generated by the **route-server-test lookup** action and summarizes the results. The results of the analysis are outputted to a specified analysis.xml file. If you do not specify an analysis.xml file, the OS-E writes the resulting output to the screen. This action allows you to view how various resources are utilized with the current routing configuration being tested. The output of the analysis has the following format.

```

Analysis of results file : /tmp/results.xml
      Analysis created on : 13:09:56.729762 Mon 2010-11-01
        Total test suites : 1
          Total test cases : 3
            Total results with routes : 3
              Total results without routes : 0
                Smallest hunt result : 1
                  Largest hunt result : 1

Route position : 1
  Total : 3
-----
Carrier "default" referenced 3 times

```

```
|--- Endpoint "example.net" referenced 3 times
```

```
Route "route-plan:2" referenced 3 times
```

```
Queries with no route results:
```

```
-----  
None
```

The **route-server-test compare** [*test-results-file1*] [*test-results-file2*] [*diff-results*] action compares two results files and outputs the differences to a specified diff-results.xml file. If you do not specify a diff-results.xml file, the OS-E writes the output to the screen. The output of the comparison has the following format.

```
Comparing results from file /tmp/results.xml with /tmp/results2.xml  
-----
```

```
Test suite 1 "DID 1000-1002"  
  |--- Query "1001"  
    |--- Route 0  
      |--- name "route-plan:2" not equal "route-plan:3"
```

The **route-server-test validate** [*results.xml*] [*output.xml*] [*table*] action compares the results file with the active routes and outputs the differences to a specified validation-results.xml file. If you do not specify a validation-results.xml file, the OS-E writes the output to the screen. Any differences between the results.xml file and the active routing tables has the following format.

```
Comparing results from file /tmp/results.xml with /tmp/results2.xml  
-----
```

```
Test suite 1 "DID 1000-1002"  
  |--- Query "1001"  
    |--- Route 0  
      |--- name "route-plan:2" not equal "route-plan:3"
```

You can also execute this action with the optional **table** parameter. This allows you to execute the validation in a different routing table other than the currently active one.

Loading Route Server Tables Without Activating Them

The **route-server** action has been enhanced to support loading route server tables into memory without activating them.

The **route-server load** action allows you to load a route.xml file into a temporary non-active routing-table that can be referenced with the *table* name.

```
route-server load <file> <table>
```

The **route-server drop** action allows you to remove a previously loaded routing table from memory.

```
route-server drop <table>
```

The existing **route-server lookup** action has been enhanced to support two new parameters. You can now specify the table loaded into memory, as well as a time parameter to specify a query in the future in order to test time of day routing.

```
route-server lookup <to-url> [from-url] [table] [time] [display-mode]
```

Viewing Route Server Statistics

The **show route-server-did** status provider allows you to display DID routes in a DID start to DID end range. This is helpful since, internally, DID routes are converted into prefix routes, making it harder for you to get a summary of active DID routes.

table	range-start	range-end	carrier	endpoint	description
active	78153000	78153999	default	example.net	Peru
active	97896000	97896999	default	example.net	Brazil

The existing **show route-server-table** status provider has been enhanced to show the table tag and route description. You can view the temporary routing-tables loaded into memory by specifying the table name. Also, a new user-defined description field and a did-entry-index has been added. When the did-entry-index is not “-1”, it can be used in conjunction with the **route-server-did** action to get more information on the DID entry.

tag	to-match	carrier	endpoint	description
active	612864*	default	example.net	Peru
active	8621289*	default	example.net	Brazil

Displaying Route Server Version Information

Select **About LCR Import** from the menu bar to display information about the route server version you are running, as illustrated in the following image.

Version	Number	Branch	Date	Time
E3.7.0	67967-dev	crux-dev-lcr-import	Nov 05, 2015	11:45AM

Route Server Actions

Route Server Lookup

When route server receives a request for least-cost-routes, it does a search to get all possible routes that matches the "to" and "from" URL prefixes.

```
NNOS-E> route-server lookup to-url from-url
```

```
route-server lookup 9788972990@.com 7818972990@company.com
```

```
-----
Carrier                Endpoint                Mapping
-----
S - 10pct Mup Customer, gateway1                ANI:7819376550
S - 10pct Mup Customer, gateway10                ANI:7819376550
N - 10pct Mup Customer, gateway4
-----
Total routes: 3
```

Route Server Controlled-Update

The **route-server-controlled-update** action updates the routing definition database on the route server.

```
NNOS-E> route-server-controlled-update <file> [activate-time]
      [peer-wait-seconds]
```

- *<file>*—A properly-formatted XML file located in the OS-E /cxc directory

- *[activate-time]*—The time to update the specified file in the format *ss:hh:mm yyyy-mm-dd*
- *[peer-wait-seconds]*—The time, in seconds, to wait for a peer to respond (Min: 20, Max: 3600, default: 20)

Route Server Controlled Activate

The **route-server-controlled activate** action activates a new route set.

```
NNOS-E> route-server-controlled activate [table] [activate-time]
[peer-wait-seconds]
```

- *<table>*—The configured route table to activate
- *[activate-time]*—The time to activate the specified file in the format *ss:hh:mm yyyy-mm-dd*
- *[peer-wait-seconds]*—The time, in seconds, to wait for a peer to respond (Min: 20, Max: 3600, default: 20)

Route Server Controlled Delete-Backup

The **route-server-controlled delete-backup** command deletes routing XML files from the backup OS-E.

```
NNOS-E> route-server-controlled delete-backup <backup-name>
```

- *[backup-name]*—The name of the backup file to delete

Route Server Controlled Cancel

The **route-server-controlled cancel** action cancels an update or activate action.

```
NNOS-E> route-server-controlled cancel [table] [peer-wait-seconds]
```

- *<table>*—The configured route table to cancel
- *[peer-wait-seconds]*—The time, in seconds, to wait for a peer to respond (Min: 20, Max: 3600, default: 20)

Execute the **show route-server-controlled-action-status** command to view all route server updates or activations that have been cancelled.

Route Server flush

The **route-server flush** action clears the route server routing table. To repopulate the route server routing table, access the route server import client and use the **Update LCR Server** function to send the latest call rate data to the cluster master.

```
NNOS-E> route-server flush
```

Route Server Status

route-server-routing-table

The **route-server-table** status provider shows the current entries in the route server's routing definition database. For example,

```
NNOS-E> show route-server-table
```

```
-----
Rate      Priority      Carrier      Data
-----
0.0066    100 N - 10pct Mup Customer  Index: 4
                                         Match: 978896!*
                                         FromMatch:
                                         Outbound: gateway4
                                         MinDuration: 6
                                         BillingIncrement: 30
                                         New ANI: sip:7819376550@.com
                                         New pAssert: sip:7819376550@covergen.com
                                         Effective Start: 00:00:00 Mon 2007-04-25
                                         Effective End: 00:00:00 Fri 2008-04-25

0.0066    100 S - 10pct Mup Customer  Index: 6
                                         Match: 97889751!*
                                         FromMatch:
                                         Outbound: gateway16
                                         MinDuration: 6
                                         BillingIncrement: 60
                                         New ANI: sip:7819376550@.com
                                         Effective Start: 00:00:00 Mon 2007-04-25
                                         Effective End: 00:00:00 Fri 2008-04-25
-----
```

route-server-controlled-action-status

The **route-server-controlled-action-status** action displays the status of route server actions such as **route-server-controlled-update**, **route-server-controlled activate**, and **route-server-controlled cancel**. For example:

```
NNOS-E> show route-server-controlled-action-status
    box: 1
    master: true
    state: Inactive
    entries: 22
```

```

    routes: 75
    load-set: policy.xml
    tableName: default

```

route-server-action-status

The **route-server-action-status** action shows the status of the **route-server flush** action. For example,

```
NNOS-E> show route-server-action-status
```

```

    status: route-server flush
    start: 16:56:21 Mon 2015-11-26
    end: 16:56:29 Mon 2015-11-26
    entries: 600000
    result: Success!

```

route-server-carriers

The **route-server-carriers** action displays the carrier name and the total number call rate entries associated with each carrier in the route server.

```
NNOS-E> show route-server-carriers
```

```

-----
Carrier          Total Entries
-----
Alltel           1391220
Comcast          834732
PacBell          695610
Sprint           139122
Tmobile          139122
Verizon          139122
Vonage           139122
-----

```

Performing Route Server Queries

You can configure the OS-E to perform multiple route server queries. In releases prior to 3.6.0m5, you could query route servers based, only, on the SIP To URI. You can now query route server based on the To, From, or Request URI or using a named variable.

To do this you first must create a **route-server-sequence** where you configure the various queries. Then under the **session-config > authorization** object you select the **sequence** you want to link to that session config.

To configure queries:

1. Select the **Configuration** tab and click on the **vsp > route-server-config** object.
2. Click **Add route-server-sequence**.
3. Specify a **name** for the route server sequence you are creating. Click **Create**.
4. Click **Add query**.
5. Specify a **name** for the query you are creating. Click **Create**. The **query** configuration object appears.

The screenshot shows the configuration interface for a query. The breadcrumb path is **vsp/route-server-config/route-server-sequence rs_sequence1/query rs_query1**. The configuration fields are as follows:

* name	rs_query1
description	
query	* type: header (Use the value of the specified SIP header) * source: Request expression: (regular expression) replacement:
table	default
lookup-type	route (Execute a route lookup.)
append	merge (Merge existing routes with new ones.)
variable-load	none (No variables expected or do not load any that may have been returned.)
variable-mappings	Add variable-mappings
variable.ignore-additional	false
condition-list	Configure
abort-on-failure	false
stop-on-success	false

6. **description**—Provide a brief description of this query.
7. **query**—Provide the following criteria for this query:
 - **type**—Specify the source of the query. This can either be a value from the **header** or a **variable**. The default setting is **header**.
 - **source**—Specify the source of the data to query. When the data type is **header**, the source can be either **Request**, **To**, or **From**. When the data type is **variable**, enter a string for this value. The default setting is **Request**.

- **expression**—When query **type** is **variable**, specify the regular expression to apply against the source value. The resulting value is what is queried.
 - **replacement**—When query **type** is **variable**, specify the value for the OS-E to use in the route server query that is derived from the **expression**.
8. **table**—Specify the table to use for this query. This is a table configured under the **route-server > table-config** object. The default setting is the **default** table.
 9. **lookup-type**—Specify the type of route server lookup.
 - **route**—Execute a route lookup.
 - **variable**—Execute a lookup on a named variable only and do not query routes.
 10. **append**—Specify how results of multiple queries should be appended.
 - **merge**—Routes from previous route server sequence queries are merged with the current query resulting routes.
 - **replace**—Routes from previous route server sequence queries are replaced with the current query resulting routes.
 11. **variable-load**—Specifies whether or not to assign route server variables to internal named variables.
 - **none**—No variables are expected or, however, if any are returned from the route server the OS-E does not load them.
 - **all**—Load all variables returned by the route server as-is into session-config named variables.
 - **assign**—Assign the specified variables returned by the route server into the specified session named variables.
 12. **variable-ignore-additional**—Specifies whether or not to ignore any additional variables that may have been returned. The default setting is **false**.
 13. **abort-on-failure**—Specifies the OS-E's behavior when a query fails. When **true**, the OS-E stops querying the route server. The default setting is **false**.
 14. **stop-on-success**—Specifies the OS-E's behavior when a query is successful. When **true**, the OS-E stops querying the route server. The default setting is **false**.

- Click **Add variable-mappings** next to **variable-mappings**. This object configures a list of mappings between route server variables returned and session-config named variables.

The screenshot shows the 'acme packet' Configuration interface. The breadcrumb trail is: Home > Configuration > Status > Call Logs > Event Logs > Actions > Services > Keys > Access > Tools. The main heading is 'Create vsplroute-server-config/route-server-sequence rs_sequence1/query rs_query1/variable-mappings - Step 1 of 1: Edit variable-mappings'. Below this, there is a text prompt: 'Please provide some basic information for variable-mappings. Then press "Create".' There are two input fields: '* route-server-variable' and '* session-named-variable'. At the bottom, there are 'Create', 'Reset', and 'Cancel' buttons.

- route-server-variable**—Specify a variable name returned by the route server.
- session-named-variable**—Specify the session-config named variable to which you want to assign the route server variable.
- Click **Create**. You are returned to the **query** object.
- Click **Configure** next to **condition-list**. This object defines the conditions required for the OS-E to execute a query.

The screenshot shows the 'acme packet' Configuration interface. The breadcrumb trail is: Home > Configuration > Status > Call Logs > Event Logs > Actions > Services > Keys > Access > Tools. The main heading is 'Configure vsplroute-server-config/route-server-sequence rs_sequence1/query rs_query1/condition-list'. Below this, there are 'Set', 'Reset', 'Back', and 'Delete' buttons. The main content area contains a table of configuration options:

operation	AND
mode	evaluate (The Net-Net OS-E runs the conditions to determine whether to apply session configuration settings.)
sip-message-condition	Add sip-message-condition
from-uri-condition	Add from-uri-condition
to-uri-condition	Add to-uri-condition
request-uri-condition	Add request-uri-condition
from-server-condition	Add from-server-condition
date-time-condition	Add date-time-condition
user-group-condition	Add user-group-condition
named-variable-condition	Add named-variable-condition
action-condition	none (not an action)

20. **operation**—Specifies the decision operation to use (AND/OR) should a condition match occur in the SIP call session.

When the **AND** operation is selected, all conditions must match for this query to be executed. If the **OR** operation is selected, a single condition match is sufficient for a query to be executed.

21. **mode**—Sets how the OS-E applies the condition list. When set to **evaluate**, the OS-E runs the conditions to determine whether or not to apply the session configuration settings. When set to **always-true**, the OS-E applies the session configuration settings, no conditions need to be configured.
22. **sip-message-condition**—See SIP Message Condition Options in the Oracle Communications OS-E Objects and Properties Reference Guide.
23. **from-uri-condition**—See From, To, and Request URI Condition Options in the Oracle Communications OS-E Objects and Properties Reference Guide.
24. **to-uri-condition**—See From, To, and Request URI Condition Options in the Oracle Communications OS-E Objects and Properties Reference Guide.
25. **request-uri-condition**—See From, To, and Request URI Condition Options in the Oracle Communications OS-E Objects and Properties Reference Guide.
26. **from-server-condition**—Specifies the criteria against which the SIP server that sent the SIP message is compared to match this policy rule.
27. **date-time-condition**—See Date and Time Condition Options in the Oracle Communications OS-E Objects and Properties Reference Guide.
28. **user-group-condition**—Specifies the user group names in the To: and From: fields to match in this policy rule. The match indicates that the SIP message caller (From:) and the recipient (To:) are members of the specified group.
29. **named-variable-condition**—Specifies the named variable to match this policy rule.
30. **action-condition**—Specifies whether the configured rule is applied to normal SIP traffic or to a specific action. Select **none** to apply the rule to SIP traffic or select either: **none**, **call-control**, **presence-subscribe**, or **presence-end-subscription**.
31. Click **Set**. Update and save the configuration.

Once you have configured your queries under the **route-server-sequence** object, you must link them to a session config.

To link a sequence to a session config:

1. Select the **Configuration** tab and click the **vsp** object.
2. Click either the **default-session-config** or **session-config-pool > entry** object.
3. Click the **authorization** object.
4. **sequence**—Select the configured **route-server-sequence** you want to link to this session config. If you have not configured a **route-server-sequence**, click **Create** next to the **sequence** property.
5. Click **Set**. Update and save the configuration.

Three status providers have been created that allow you to view information regarding multi-stage routing.

The **show route-server-table-config** action displays information regarding tables configured under the **route-server > table-config** object.

```
NNOS-E>show route-server-table-config
```

```
name          filename      description          routes
-----
default       routes.xml    My default routes    1000
lerg6         lerg6.xml     Telecordia exchanges 300000
```

Field	Description
name	Name of the tagged table.
filename	Name of the table's associated route file.
description	Description of the table.
routes	Number of routes associated with the table.

The **show route-server-sequence** action displays information regarding sequences configured under the **route-server-sequence** object.

```
NNOS-E>show route-server-sequences
```

```
name          description          hits
-----
Coverage     Use Coverage specific queries    888
Acme         Use Acme specific queries        1
```

Field	Description
name	Name of the route server sequence.

Field	Description
description	Description of the route server sequence.
routes	Number of times the sequence was referenced.

The **show route-server-query** action displays information regarding queries configured under the **route-server-sequence** object.

```
NNOS-E>show route-server-queries
```

```
sequence      query          type           description                    hits
-----      -
Coverage     cgnLERN        variable       query the calling number OCN/LATA 10
Coverage     cdnLERN        variable       query the called number OCN/LATA 10
Coverage     interLATA      route          query inter-LATA routes          6
Coverage     intraLATA      route          query intra-LATA routes          4
```

Field	Description
sequence	Name of the route server sequence.
query	Name of the route server query.
type	Type of query. This can be either route or variable.
description	Description of the route server query.
hits	Number of times the query was referenced.

Viewing This Document

This document is Chapter 12 of the *Net-Net OS-E – Session Services Configuration Guide*.

You can view this chapter (LCR.pdf) by selecting **Help** from the route server menu bar from any screen. This launches Adobe Acrobat (if installed on your PC) or Adobe Reader (available with most PCs today). Go to www.adobe.com if you need the free Adobe Reader download.

Chapter 10. Admission Control

About This Chapter

This chapter covers OS-E admission control for SIP INVITE, SIP REGISTER, and TLS sessions. Admission control allows you to limit calls to the OS-E that might otherwise heavily consume OS-E memory and storage resources.

For detailed descriptions of the configuration properties covered in this chapter, refer to the Oracle Communications OS-E Objects and Properties Reference Guide.

Call Admission Control

OS-E call admission control (CAC) allows you to control or limit the number of calls to and from various SIP devices. Call admission control, which only applies to SIP INVITE traffic, operates at the following levels:

- VSP (operating on all calls to the OS-E system.)
- SIP gateways (carrier and enterprise)
- Trunk groups
- Calling groups
- User agents (location)

Call admission control prevents malfunctioning or improperly-configured devices from consuming critical resources that impact the performance of the OS-E system. Call routing loops, for example, can be prevented with call admission control since it is not always possible for the OS-E to detect certain types of loops (per RFC 3261) when the next-hop device is a back-to-back user agent (B2BUA).

VSP Control

By default, VSP call admission control is *disabled*. When *enabled*, you can limit the total number of calls that can be processed by the OS-E at any time. Use the **call-admission-control** property to enable or disable call admission control.

CLI Session

```
config> config vsp admission-control
config admission-control> set call-admission-control enabled
```

Call control limits are optimized for the specific platform on which you are running the OS-E with a default setting of *automatic*. You can display the automatic values with the **show automatic-settings** status provider.

```
NNOS-E> show automatic-settings
```

name	value
----	-----
cac-max-calls	7500
cac-max-calls-in-setup	1500
cac-max-number-of-tls	3000
cac-max-tls-in-setup	425
cac-min-calls-in-setup	10
max-number-of-sessions	7500
max-routes	1048576
stack-socket-event-threads-max	4
stack-socket-threads-max	4
stack-worker-threads	4

The current settings can be viewed with the **show call-admission-control** command.

```
NNOS-E> show call-admission-control
```

name:	default
call-admission-control:	disabled
max-calls:	7500
max-calls-in-setup:	1500
min-calls-in-setup:	10
calls-in-setup-dynamic-threshold:	1500
cpu-monitor-span:	20 seconds
cpu-monitor-interval:	10 seconds
average-sip-cpu:	0 %
calls-high-cpu-threshold:	90 %
calls-low-cpu-threshold:	50 %
current-calls:	0
current-calls-in-setup:	0
most-calls:	0
most-calls-in-setup:	0

```
max-calls-dropped: 0
max-calls-dropped-last-logging:
max-calls-in-setup-dropped-this-interval: 0
max-calls-in-setup-dropped-last-interval: 0
max-calls-in-setup-dropped: 0
```

Limiting Based On Calls

The following VSP **admission-control** settings restrict the number of calls that can be processed by the VSP.

- **cac-max-calls**—The maximum number of allowed concurrent calls.
- **cac-max-calls-in-setup**—The maximum number of allowed calls in the setup stage.

Limiting Based On CPU

The following VSP settings calculate a dynamic threshold so that the OS-E rejects calls based on the CPU usage. The initial dynamic threshold value is the **cac-max-calls-in-setup** setting.

- **cpu-monitor-span**—The number of seconds over which the OS-E calculates the total system CPU average. At the end of the span, the average value is compared to the call CPU thresholds to determine whether to modify the dynamic threshold. The longer the span, the fewer the changes to the thresholds. A shorter span will result in reaction to brief CPU activity spikes.
- **cpu-monitor-interval**—The frequency in seconds over which the OS-E calculates the total system CPU average for the last span.
- **calls-high-cpu-threshold**—When this threshold is reached, the dynamic threshold value decreases by 10% but never goes below the **cac-min-calls-in-setup** setting.
- **calls-low-cpu-threshold**—When this threshold is reached, the dynamic threshold value increases by 16% if the average CPU is less than the low threshold and by 4% if the less than the high threshold.
- **cac-min-calls-in-setup**—This lowest possible value of the dynamic threshold.
- **call-response-code-at-threshold**—The response code sent when a request was rejected because the dynamic threshold was reached.
- **call-response-string-at-threshold**—The response string sent when a request was rejected because the dynamic threshold was reached.

Server Control

You can configure the OS-E to perform call admission control under the following server objects:

- **vsp/enterprise/servers/sip-gateway** *name*/**server-pool**/*server name*
- **vsp/carriers/carrier**/*gateway name*
- **vsp/carriers/carrier**/*gateway name*/**trunk-group** *name*
- **vsp/calling-group**/*group name*

Each of these server objects can be set to limit inbound or outbound calls based on absolute values or estimated bandwidth. The OS-E currently keeps track of inbound and outbound calls using the same counters, so inbound calls may affect emission control and the reverse.

You can enable or disable admission (inbound) or emission (outbound) calls with the respective server object properties. You can view the current settings with the following commands:

- **show sip-server-cac**
- **show gateway-cac**
- **show trunk-cac**
- **show calling-group-cac**

Limiting Based On Calls

The following settings restrict the amount of calls that can be sent or received from a server.

- **max-number-of-concurrent**—The maximum number of allowed concurrent calls.
- **max-calls-in-setup**—The maximum number of allowed calls in the setup stage.
- **call-rate-limiting** (secondary)—The number of calls allowed during a given period of time. For example, if the *calls-per-interval* setting is 60 and the *smoothing-interval* setting is 1, the OS-E allows 60 calls/second. Once that limit is reached, the OS-E attempts to hunt for another server. If no servers are found, the OS-E rejects the call with the specified *result-code* and *result-string*.
 - *calls-per-interval*—The maximum number of calls allowed in this period.

- *smoothing-interval*—The period when the OS-E allows a burst of calls without rejection.
- *result-code*—The response code sent when a request is rejected.
- *result-string*—The response string sent when a request is rejected.

Limiting Based On Bandwidth

The following setting restricts the bandwidth to and from a server. The bandwidth is an estimate based on the CODEC negotiated in the SDP. You can view the estimated CODEC bandwidth with the **show codec-info** status command.

- **max-bandwidth**—The maximum bandwidth that can handled by this server. When set to unlimited (default), the bandwidth is limited only by the physical links or processing engine.

User-Agent Control

You can configure the OS-E to perform call admission control for individual User-Agents (UAs). These settings are found under the **session-config/location-call-admission-control** configuration object.

When a UA first registers, the values are copied to the location-cache entry. The session configuration can be from either the default-session-config or any session-config entry pools that are associated with this UA during the registration process. You can view the current setting with the **show location-cache-cac** status command.

- **max-number-of-concurrent**—The maximum number of allowed concurrent calls.
- **max-calls-in-setup**—The maximum number of allowed calls in the setup stage.
- **call-rate-limiting**—The number of calls allowed in a certain period of time. For example, if the *calls-per-interval* setting is 60 and the *smoothing-interval* setting is 1, the OS-E allows 60 calls/second. Once that limit is reached, the OS-E attempts to hunt for another server. If no servers are found, the OS-E rejects the call with the specified *result-code* and *result-string*.
 - *calls-per-interval*—The maximum number of calls allowed in this period.
 - *smoothing-interval*—The period for which we allow a burst of calls without rejecting.

- *result-code*—The response code sent when a request is rejected.
- *result-string*—The response string sent when a request is rejected.

Limiting Based On Bandwidth

The following setting restricts the bandwidth to and from a UA. The bandwidth is an estimate based on the CODEC negotiated in the SDP. You can view the estimated CODEC bandwidth with the **show codec-info** status command.

- **max-bandwidth**—The maximum bandwidth that can handled by this UA. When set to unlimited (default), the bandwidth is limited only by the physical links or processing engine.

Using the Session-Config Override

The OS-E call-admission-control feature is bypassed when the emergency-settings feature in the session-config is enabled. This allows the administrator to associate this session-config to a an emergency dial-plan (for example 911) or to dynamically load it on the session using RADIUS or WSDL when placing a call. These calls still count towards future CAC checks but would not be rejected.

Registration Admission Control

The OS-E registration admission control feature allows you to control or limit the amount of registration to a server. Registration admission control, which applies to SIP REGISTERS, operates at the following levels:

- VSP (operating on all calls to the OS-E system)
- SIP gateways (enterprise)

Registration admission control, when enabled, prevents the OS-E and/or the server from accepting more registrations than it can possibly process.

VSP Control

By default, VSP registration admission control is disabled. When enabled, you can limit the total number of registrations that can be processed by the OS-E at any time. Use the **registration-admission-control** property to enable or disable registration admission control.

CLI Session

```
config> config vsp admission-control
config admission-control> set registration-admission-control enabled
```

Registration control limits are optimized for the specific platform on which you are running the OS-E with a default setting of *automatic*. You can display the automatic values with the **show automatic-settings** status provider.

```
NNOS-E> show automatic-settings

name                               value
----                               -
cac-max-calls                       7500
cac-max-calls-in-setup              1500
cac-max-number-of-tls               3000
cac-max-tls-in-setup                425
cac-min-calls-in-setup              10
max-number-of-sessions              7500
max-routes                           1048576
stack-socket-event-threads-max      4
stack-socket-threads-max            4
stack-worker-threads                 4
```

The current settings can be viewed with the **show registration-admission-control** command.

```
NNOS-E> show registration-admission-control

                                name: default
                                registration-admission-control: disabled
                                max-registrations: 30000
                                pending-registrations-high-watermark: 500
                                pending-registrations-low-watermark: 10
                                pending-registrations-dynamic-threshold: 500
                                cpu-monitor-span: 20 seconds
                                cpu-monitor-interval: 10 seconds
                                average-sip-cpu: 0 %
                                registrations-high-cpu-threshold: 90 %
                                registrations-low-cpu-threshold: 70 %
                                total-client-bindings: 0
```

```
registrations-in-progress: 0
registrations-most-in-progress: 0
registrations-sessions: 0
processed-new-registrations: 0
processed-waiting-registrations: 0
processed-challenged-registrations: 0
processed-other-registrations: 0
suppressed-registrations-this-interval: 0
suppressed-registrations-last-interval: 0
suppressed-new-registrations: 0
suppressed-waiting-registrations: 0
suppressed-challenged-registrations: 0
last-register-suppressed-at:
discarded-other-registrations: 0
last-register-discarded-at:
edp-transactions-in-progress: 0
```

Limiting Based On Registers

The following setting restricts the amount of registers that can be processed by the VSP.

- **max-number-of-registrations**—The total number of registrations that can be processed by this VSP.

Limiting Based On CPU

The following settings calculate a dynamic threshold so that the OS-E rejects calls based on the CPU usage. The initial dynamic threshold value is the **pending-registrations-high-watermark**.

- **cpu-monitor-span**—The number of seconds over which the OS-E calculates the total system CPU average. At the end of the span, the average value is compared to the call CPU thresholds to determine whether to modify the dynamic threshold. The longer the span, the fewer the changes to the thresholds. A shorter span will result in reaction to brief CPU activity spikes.
- **cpu-monitor-interval**—The frequency in seconds over which the OS-E calculates the total system CPU average for the last span.
- **registrations-high-cpu-threshold**—When this threshold is reached, the dynamic threshold value decreases by 10% but never goes below **pending-registrations-low-watermark**.

- **registrations-low-cpu-threshold**—When this threshold is reached, the dynamic threshold value increases by 16% if the average CPU is less than the low threshold and by 4% if the less than the high threshold.
- **pending-registrations-low-watermark**—This lowest possible value of the dynamic threshold.
- **pending-registrations-low-watermark**—This highest possible value of the dynamic threshold.

Server Control

You can configure the OS-E to perform registration admission control under the `vsp\enterprise\servers\sip-gateway name\server-pool\server name`

This object can be set to an absolute number of registrations that it can accept. The admission-control setting on the server or gateway must be set to enabled for registration-admission-control to be active.

You can view the current settings with the **show sip-server-cac** and **show gateway-cac** status commands.

Limiting based on registers

The following settings restrict the amount of registers that can be processed by this server.

- **max-number-of-registrations**—The total number of registrations that this server can handle.
- **max-registrations-in-progress**—The total number of registrations in progress that this server can handle.

TLS Admission Control

The OS-E TLS admission control allows you to control or limit the amount of TLS connections that can be established. The control is applied at the VSP configuration level only.

VSP Control

By default, VSP TLS admission control is disabled. When enabled, you can limit the total number of TLS connections that can be processed by the OS-E at any time using the VSP admission-control object.

- **cac-max-number-of-tls**—The total number of TLS connections that can be established.
- **cac-max-tls-in-setup**—The total number of TLS connections in progress.

TLS admission control limits are optimized for the specific platform on which you are running the OS-E with a default setting of *automatic*. You can display the automatic values with the **show automatic-settings** status command.

```

NNOS-E> show automatic-settings
name                               value
----                               -
cac-max-calls                       7500
cac-max-calls-in-setup               1500
cac-max-number-of-tls                3000
cac-max-tls-in-setup                 425
cac-min-calls-in-setup               10
max-number-of-sessions               7500
max-routes                           1048576
stack-socket-event-threads-max      4
stack-socket-threads-max             4
stack-worker-threads                 4

```

The current settings can be viewed with the **show tls-admission-control** status command.

```

NNOS-E> show tls-admission-control
name: default
tls-admission-control: disabled
max-calls: 7500
max-calls-in-setup: 1500
current-calls: 0
current-calls-in-setup: 0
max-calls-dropped: 0
max-calls-in-setup-dropped: 0

```

Chapter 11. Calling Groups

About This Chapter

This chapter covers the OS-E calling group configuration. A calling group allows you to group phones behind a single device such as a PBX or an ATA where the associated lines do not have registration capabilities. The **calling-group** object creates different groups (profiles) that can be referenced through the **registration-plan**, **dial-plan**, and session config **calling-group-settings** objects to control routing of outgoing calls.

It creates a way to segregate routing arbitration, call routing, policy, and normalization based on the user group

For detailed descriptions of the configuration properties covered in this chapter, refer to the Oracle Communications OS-E Objects and Properties Reference Guide.

Calling Groups Overview

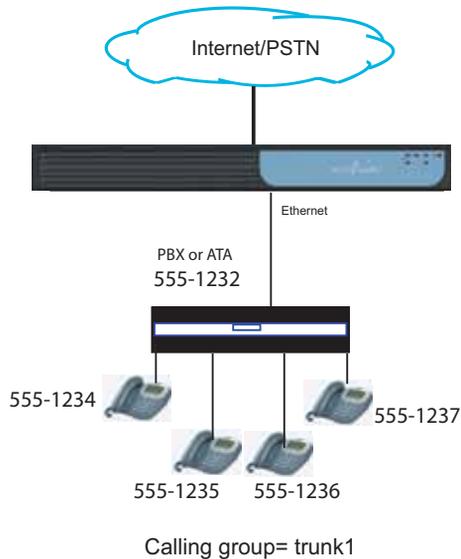
Calling groups are useful when an endpoint is on a dynamic IP address, such as a DSL circuit or shared port adapter. When configured, the OS-E learns the address dynamically (from the incoming REGISTER request) and uses routes configured specifically for that group instead of being forwarded into the general dial-plan table. Calling groups are intended for devices that shelter multiple endpoints behind them.

You must enable **calling-group-routing** in the VSP **settings** object for dynamic learning and the route and source-route functionality. Calling group routing works as follows.

1. The OS-E receives a REGISTER request.
2. If the REGISTER matches a configured **registration-plan**, the OS-E checks to see if there is an associated calling-group.

3. If there is an associated calling-group, and if **calling-group-routing** is enabled, the OS-E binds the calling group to the IP address of the device that sent the registration.
4. A calling group can have only one associated IP address. If the OS-E receives a REGISTER for an existing calling group but the IP address is different, it overwrites the known address with the new one.
5. When the OS-E receives an INVITE, and **calling-group-routing** is enabled, it checks the IP address of the endpoint against all configured calling groups. If a match exists, the OS-E performs a **dial-plan** lookup within the configured calling-group routes and source-routes. If a match does not exist, the call is not routed. If **calling-group-routing** is disabled, the OS-E uses the routes and source routes within dial-plan to handle the call.

You can also direct incoming calls to a calling group. To do so, set the dial-plan\route **peer** property to **calling-group** and reference the group through which you want matching calls routed.



Configuring Calling Groups

Configuring a basic calling-group configuration consists of the steps below. In this example, the registration point is placed into the calling group by the registration-plan, and any calls that match the dial-plan will be sent to that calling-group.

1. Create a static directory of non-contiguous users of the calling group using the `vsp\enterprise\directories` object.



Note: If the numbers in the SIP trunk are contiguous as in the image above, then you do not need to create a directory. Use a domain, prefix, or other common matching criteria for the dial-plan and registration-plan.

```
NNOS-E> config vsp enterprise directories
vsp enterprise directories> config static-directory Trunk1
config static-directory> set tag Trunk1
config static-directory> config user-attributes
config user-attributes> set name Trunk1
config user-attributes> return
config static-directory> set domain domain.com
```

```
config static-directory> config user 5551234
Creating 'user 5551234'
config user 5551234> set address sip:5551234@domain.com
config user 5551234> config attribute Trunk1
Creating 'attribute Trunk1'
config attribute Trunk1> set value 5551234
config attribute Trunk1> return
config user 5551234> return
```

```
config static-directory> config user 987654321
Creating 'user 987654321'
config user 987654321> set address sip:987654321@domain.com
config user 987654321> config attribute Trunk1
config attribute Trunk1> set value 987654321
config attribute Trunk1> return
```

2. Create the **calling-group**.

```
NNOS-E> config vsp
config vsp> config calling-groups
config calling-groups> config group Trunk1
config group Trunk1> set admin enabled
config group Trunk1> set domain domain.com
config group Trunk1> set routing-tag Trunk1
```

3. Add a **registration-plan** that maps to the calling-group. You would need to either Accept or Delegate the registration.

```
NNOS-E> config vsp registration-plan
config registration-plan> config route Trunk1
Creating 'route Trunk1'
config route Trunk1> set to-uri-match directory
    "vsp\enterprise\directories\static-directory Trunk1"
config route Trunk1> set action accept
config route Trunk1> set calling-group "vsp\calling-groups\group
    Trunk1"
Creating 'vsp\calling-groups\group Trunk1'
```

4. Add a **dial-plan** that sends calls to the users behind the UA/PBX back to the calling-group.

```
NNOS-E> config vsp dial-plan
config dial-plan> config route Trunk1
config route Trunk1> set request-uri-match directory
    "vsp\enterprise\directories\static-directory Trunk1"
config route Trunk1> set location-match-preferred exclusive
config route Trunk1> set peer calling-group "vsp\calling-groups\group
    Trunk1"
config route Trunk1>
```

Chapter 12. Net-Net OS-E SIP Trunking

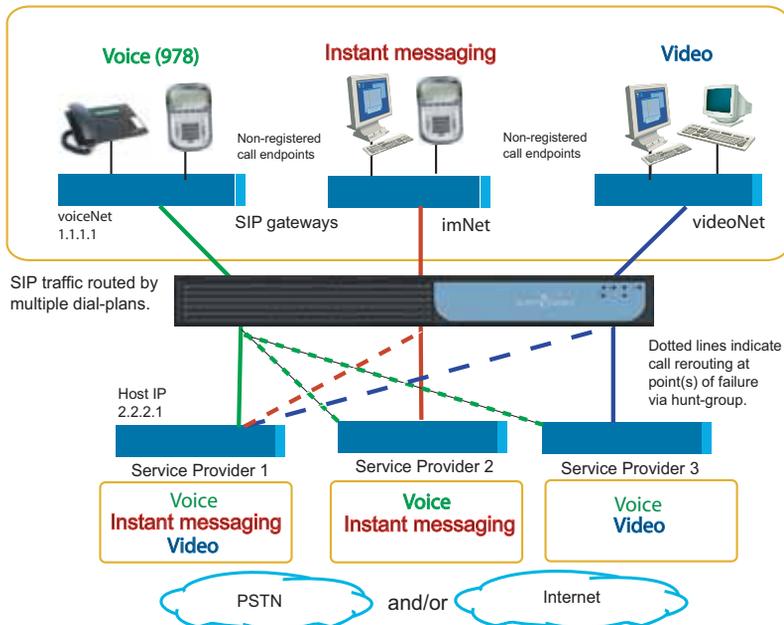
About This Chapter

This chapter provides information on configuring SIP trunking. A SIP trunk is a server-to-server network that allows enterprises to securely interconnect “islands” of collaboration, such as voice, video, and instant messaging systems within the enterprise and with Telco (service provider) partners. In the event of a line failure at some critical point in the enterprise, SIP trunks reliably reroute calls around the point of failure to their intended destinations.

Sample SIP Trunking Network

The following image illustrates a sample SIP trunking network with connected islands of collaboration: voice, video, and instant messaging. Each “island” consists of non-registered call endpoints that are accessible using separate OS-E dial plans. Hunt groups control call rerouting in the event of a failure to secondary service providers.

Connected “islands” of collaboration on enterprise LAN



Configuring a SIP Trunk

There are several steps necessary to configuring a SIP trunk.

1. Configure the enterprise **SIP gateways** at each “island” of collaboration to direct SIP traffic inbound to the OS-E, outbound to the intended carrier.
2. Configure each **carrier**, where service providers direct traffic *inbound* to the OS-E, and *outbound* to the intended “island” of collaboration.
3. Each carrier requires a **hunt-group** configuration that contains a prioritized list of service providers to be accessed in the event of a failure. For example, in the event of a failure on the voice network to Service Provider 1, Service Provider 2 and Service Provider 3 are alternate voice carriers.

4. Configure a **dial-plan** for the “islands” of collaboration and for each **carrier**. In this network, you will need three unique dial plan routes to support the voice, instant messaging, and video network. Each route defines the destination to the appropriate service provider. Similarly, you will need to configure a dial-plan for each carrier to direct traffic back to the “islands” of collaboration.

Configure the Enterprise SIP Gateway

The following CLI session configures the voiceNet enterprise SIP gateway as in the image above. SIP call traffic from phones connected to this gateway will generate inbound sessions to the OS-E before the traffic is directed outbound to the destination carrier. The **sip-gateway** configuration uses session configuration pool entries that direct these inbound and outbound sessions from the configured peers.

On the VoiceNet Enterprise SIP Gateway

```
CXC> config vsp enterprise servers
config servers> config sip-gateway voiceNet
Creating 'sip-gateway voiceNet'
config sip-gateway voiceNet> set local 1.1.1.1
config sip-gateway voiceNet> set domain company.com
```

Configure the Carrier Network

The following CLI session configures the Service Provider_1 carrier as in the figure above. The carrier configuration is similar to SIP enterprise gateway. Calls matching the carrier IP address will be routed inbound to the OS-E and then outbound to the voiceNet peer.

For ServiceProvider_1

```
CXC> config vsp carriers
config carriers> config carrier ServiceProvider_1
Creating 'carrier ServiceProvider_1'
config carrier ServiceProvider_1> set admin enabled
config carrier ServiceProvider_1> set description verizon
config carrier ServiceProvider_1> set carrier verizonMaynard

config carrier ServiceProvider_1> config exchange Maynard
config exchange Maynard> config switch 2.2.2.1
config switch 2.2.2.1> set host 2.2.2.1
```

Create the Hunt-Group

The following CLI session creates a **hunt-group** for ServiceProvider_1. The **hunt-group** lists the providers to hit in sequence should the preferred provider(s) become unavailable due to an outage. In the **hunt-group** for ServiceProvider_1 supporting the voice network (see the image above), note that ServiceProvider_2 and ServiceProvider_3 carriers also support voice applications, and are listed as options.

```
config carriers> config hunt-group ServiceProvider_1
Creating 'hunt-group ServiceProvider_1'
config hunt-group ServiceProvider_1> set admin enabled

config hunt-group ServiceProvider_1> set option carrier "vsp carriers
carrier ServiceProvider_1 exchange Maynard switch 2.2.2.1
config hunt-group ServiceProvider_1> set option carrier "vsp carriers
carrier ServiceProvider_2"
config hunt-group ServiceProvider_1> set option carrier "vsp carriers
carrier ServiceProvider_3"
config hunt-group ServiceProvider_1> show
vsp
carriers
  hunt-group ServiceProvider_1
  admin enabled
  option[1] carrier "vsp\carriers\carrier ServiceProvider_1\exchange
  Maynard\switch 2.2.2.1" none
  option[2] carrier "vsp\carriers\carrier ServiceProvider_2" none
  option[3] carrier "vsp\carriers\carrier ServiceProvider_3" none
```

Configure the Enterprise Dial-Plan

The section covers the **dial-plan** configuration for SIP traffic on the voiceNet gateway as in the image above. SIP traffic matching 978 from the voiceNet gateway will be routed (forwarded) to ServiceProvider_1, covered in the next section.

For the Voice Network

```
CXC> config vsp dial-plan
config dial-plan> config route voiceNet978
Creating 'route voiceNet978'
config route voiceNet978> set admin enabled
config route voiceNet978> set action forward
config route voiceNet978> set peer hunt-group "vsp carriers hunt-group
ServiceProvider_1"
config route voiceNet978> set request-uri-match condition-list

config route voiceNet978> config condition-list
condition-list> set sip-message-condition remote-ip match 1.1.1.1/24
```

Configure the Carrier Dial-Plan

The section covers the **dial-plan** configuration for SIP traffic from the ServiceProvider_1 carrier as in the image above. SIP traffic matching the carrier IP address will be routed (forwarded) to the voiceNet enterprise SIP gateway.

For ServiceProvider_1

```
CXC> config vsp dial-plan
config dial-plan> config route toVoiceNet978
Creating 'route toVoiceNet978'
config route toVoiceNet978> set admin enabled
config route toVoiceNet978> set action forward
config route tovoiceNet978> set peer server "vsp enterprise servers
    sip-gateway voiceNet"
config route tovoiceNet978> set request-uri-match condition-list

config route voiceNet978> config condition-list
condition-list> set sip-message-condition remote-ip match 2.2.2.1/24
```

Chapter 1. WebRTC Overview

What is WebRTC?

WebRTC is an open source technology standard that enables browser to browser communications for voice, video, and P2P file sharing without the need for plugins.

WebRTC implements three JavaScript APIs:

- `getUserMedia`—Get access to data streams
- `RTCPeerConnection`—Audio or video calling with facilities for encryption and bandwidth management
- `RTCDataChannel`—Peer-to-peer communication of generic data

While WebRTC does not include any standards for signaling, the OS-E supports two types of WebRTC signaling:

- WebRTC using SIP signaling over websockets
- WebRTC using OS-E Call Control REST APIs

In addition to supporting WebRTC, the OS-E can also act as a multimedia streaming server (MSS). The MSS allows SIP and H.323 endpoints to communicate over web-based multimedia applications using either a third-party Flash media server or directly on the OS-E as an internal media server. For more information on how to configure the MSS, see the Oracle Communications OS-E Session Services guide.

WebRTC Media Handling

Like SIP endpoints, WebRTC endpoints use Session Description Protocol (SDP) as a means to exchange media capabilities. Using the WebRTC APIs, a browser can access users' cameras and microphones and transmit these media streams over the network. In order to provide secure and reliable transmission across a variety of network topologies, all WebRTC endpoints must support both Interactive Connectivity Establishment (ICE) and Secure Real-Time Transport Protocol (SRTP) in their SDP exchanges.

For more information on SDP, visit <http://tools.ietf.org/html/rfc4566>.

What is ICE?

ICE is a protocol that establishes network paths for UDP-based media streams. It is an extension of the SDP offer/answer model and works by discovering and including all possible media transport addresses (known as candidates) in the SDP. Once SDPs are exchanged, ICE tests all possible media paths using the Session Traversal Utilities for the NAT (STUN) protocol as connectivity checks. Once the connectivity checking completes, the ICE agents settle on a final candidate pair to use for media transmission. The OS-E supports ICE on a per call-leg basis, meaning it can act as both the offering and answering ICE agent to satisfy this WebRTC requirement.

In addition to ICE, the OS-E also supports augmented ICE. In ICE the OS-E strips the candidates from the SDP while in augmented ICE the OS-E preserves all candidates received from a WebRTC endpoint. This provides the WebRTC endpoints the option to either anchor media on the OS-E or not.

For more information on ICE, visit <http://tools.ietf.org/html/rfc5245>.

What is STUN?

In addition to connectivity checking, ICE relies heavily on STUN to discover all possible media candidates. During this candidate gathering phase, ICE agents perform STUN requests to discover their public IP addresses when behind a NAT device. The OS-E can be configured as a STUN server to satisfy these initial STUN requests.

For more information on STUN, visit <http://www.ietf.org/rfc/rfc3489>.

What is SDES-SRTP?

SRTP is secure RTP designed to provide encryption, authentication, and integrity to RTP streams. In SDES-SRTP, encryption keys are exchanged in the SDP offer and answer using the crypto attribute. The OS-E supports SDES-SRTP encryption and decryption on a per call-leg basis to satisfy this WebRTC requirement.

For more information on SDES, visit <http://tools.ietf.org/html/rfc4568>.

What is DTLS?

In addition to SDES-SRTP, the OS-E also supports Datagram Transport Layer Security (DTLS) as a method for encryption. DTLS works similarly to SDES-SRTP in that encryption keys are exchanged in the SDP offer and answer using the crypto attribute and the OS-E supports DTLS on a per call-leg basis.

For more information on DTLS, visit <http://tools.ietf.org/html/rfc4347>.

Configuring ICE and STUN

To configure ICE on the OS-E, you must enable session-wide media anchoring.

You must also enable symmetric RTP, which returns RTP based on the source IP address and UDP port in the received RTP. NAT modifies data in the IP header only and the SDP payload is left unchanged. By using the source IP address and UDP port from the received RTP, the OS-E sends traffic back to the NAT device instead of the untranslated addresses in the SDP.

In addition to these session-wide settings, you must also configure ICE for incoming and outgoing WebRTC sessions.

To enable system-wide media anchoring and symmetric RTP:

1. Click the **Configuration** tab and select either **default-session-config** or **session-config-pool > entry**.

2. Click **Configure** next to **media**.

The screenshot shows the Acme Packet Configuration interface. The top navigation bar includes 'Home', 'Configuration', 'Status', 'Call Logs', 'Event Logs', 'Actions', 'Services', 'Keys', 'Access', and 'Tools'. The main content area is titled 'Configure vsp|default-session-config|media'. On the left, a tree view shows the configuration hierarchy: cluster > box 1 > vsp > default-session-config > media. The main configuration area has buttons for 'Set', 'Reset', 'Back', and 'Delete'. Below these are two rows of configuration items: 'anchor' is set to 'enabled' (with a note '(media anchoring is enabled)') and 'nat-traversal' has a 'Configure' link.

3. **anchor**—Set to **enabled** to enable media anchoring for this media session. Media anchoring forces the SIP media session to traverse the OS-E.
4. Click **Configure** next to **nat-traversal**.

The screenshot shows the Acme Packet Configuration interface. The top navigation bar is the same as in the previous screenshot. The main content area is titled 'Configure vsp|default-session-config|media|nat-traversal'. On the left, the tree view shows: cluster > box 1 > vsp > default-session-config > media > nat-traversal. The main configuration area has buttons for 'Set', 'Reset', 'Back', and 'Delete'. Below these are three rows of configuration items: 'admin' is set to 'enabled' (with a note '(Resource is active)'), 'symmetricRTP' is set to 'true', and 'asymmetric-rtp-address' has an 'Edit asymmetric-rtp-address' link.

5. **symmetricRTP**—Set to **true** to enable symmetric RTP for this media session. When enabled, symmetric RTP returns RTP based on the source IP address and UDP port in the received RTP. NAT modifies data in the IP header only and the SDP payload is left unchanged.
6. Click **Set**. You are returned to the **media** object.
7. Click **Set**. Update and save the configuration.

To enable ICE for incoming WebRTC sessions:

1. Click the **Configuration** tab and select either **default-session-config** or **session-config-pool > entry**.

2. Click **Configure** next to **in-ice-settings**.

The screenshot shows the Acme Packet Configuration interface. The main navigation bar includes Home, Configuration, Status, Call Logs, Event Logs, Actions, Services, Keys, Access, and Tools. The current page is titled 'Configure vspldefault-session-configin-ice-settings'. On the left, a tree view shows the configuration hierarchy: cluster > box 1 > vsp > default-session-config > in-ice-settings. The main content area displays a table of settings for the 'in-ice-settings' resource:

Setting	Value	Notes
admin	enabled	(Resource is active)
connectivity-check-time-out	100	ms
connectivity-check-max-retransmits	200	(from 0 to 255)
delay-stun-responses	disabled	(Resource is inactive)
suppress-re-invites	disabled	(Resource is inactive)

3. **admin**—Set to **enabled** to enable ICE on this call leg.
4. **connectivity-check-max-retransmits**—Specify the number of times the OS-E retransmits ICE STUN connectivity checks before labeling a candidate pair as Failed. To achieve maximum interoperability with Chrome, set this value to no less than **200**.
5. Click **Set**. Update and save the configuration.

To enable ICE for outgoing WebRTC sessions:

1. Click the **Configuration** tab and select either **default-session-config** or **session-config-pool > entry**.
2. Click **Configure** next to **out-ice-settings**.

The screenshot shows the Acme Packet Configuration interface. The main navigation bar includes Home, Configuration, Status, Call Logs, Event Logs, Actions, Services, Keys, Access, and Tools. The current page is titled 'Configure vspldefault-session-configout-ice-settings'. On the left, a tree view shows the configuration hierarchy: cluster > box 1 > vsp > default-session-config > out-ice-settings. The main content area displays a table of settings for the 'out-ice-settings' resource:

Setting	Value	Notes
admin	enabled	(Resource is active)
connectivity-check-time-out	100	ms
connectivity-check-max-retransmits	7	(from 0 to 255)
delay-stun-responses	enabled	(Resource is active)
suppress-re-invites	enabled	(Resource is active)

3. **admin**—Set to **enabled** to enable ICE on this call leg.

4. **delay-stun-responses**—*Advanced property*. Set to **enabled**. When enabled, the OS-E does not respond to STUN until the 200 OK is received.
5. **Note:** To view Advanced properties, you must click the **Show advanced** button.
6. **suppress-re-invites**—*Advanced property*. Set to **enabled**. When enabled, the OS-E does not send a re-INVITE when ICE completes successfully.
7. Click **Set**. Update and save the configuration.

If you are configuring the OS-E for augmented ICE you must complete the configuration procedure for ICE plus some additional configuration.

To configure augmented ICE:

1. Click the **Configuration** tab and select either **default-session-config** or **session-config-pool > entry**.
2. Click **Configure** next to **media**.
3. **augmented-ice**—Set to **enabled** to enable augmented ICE.
4. Click **Set**. You are returned to the **media** object.
5. Click **Configure** next to **in-encryption**.
6. **mode**—Select **pass-thru** from the drop-down list.
7. Click **Set**. You are returned to the **media** object.
8. Click **Configure** next to **out-encryption**.
9. **mode**—Select **pass-thru** from the drop-down list.
10. Click **Set**. Update and save the configuration.

Configuring STUN

In addition to an ICE server, the OS-E can also be configured as a STUN server.

To configure the OS-E as a STUN server:

1. Click the **Configuration** tab and select the **cluster > box > interface > ip** object.

2. Click **Configure** next to **stun-server**.

The screenshot shows the acmePacket Configuration interface. The breadcrumb path is Configuration > cluster > box 1 > vsp > default-session-config > tls > static-stack-settings > session-config-pool > dial-plan > enterprise > accounting > dns > h323-settings > multimedia-streaming-config > stun-server. The 'stun-server' configuration table is as follows:

Parameter	Value	Notes
admin	enabled	(Resource is active)
port	Add port	
certificate	<input type="text"/> Create	
stun-auth-level	allow	(authentication is allowed/verified, but not required)
short-term-user-secret	<input type="text"/> Manage Password	
secondary-interface	<input type="text"/> Create	
allow-turn	enabled	(Resource is active)
relay-interface	<input type="text"/> Create	
allocation-lifetime-max	<input type="text" value="3600"/> seconds	(from 1 to 100,000, default=3600)
allocation-bandwidth-max	<input type="text" value="500"/> kbits-per-second	(from 1 to 1,000,000, default=500)
allocation-bandwidth-default	<input type="text" value="150"/> kbits-per-second	(from 1 to 1,000,000, default=150)
ta	<input type="text" value="3000"/> milliseconds	(from 0 to 10,000, default=3000)

3. **admin**—Set to **enabled** to enable the OS-E as a STUN server.

4. Click **Add port** to configure a port for the STUN server.

The screenshot shows the 'Create clusterbox 1\interface eth0\ip alstun-server\port UDP 3478 - Step 1 of 1: Edit port UDP 3478' form. The form fields are:

- * transport**: UDP (User Datagram Protocol)
- * port**: 3478 (at minimum 1, default=3478)

Buttons: [Create](#), [Reset](#), [Cancel](#)

5. **transport**—Select from the drop-down list the transport protocol over which STUN messages are exchanged between a SIP endpoint and the OS-E STUN server. Valid values are **UDP**, **TCP**, and **TLS**. The default value is UDP.
6. **port**—Specify the port over which STUN messages are exchanged between a SIP endpoint and the OS-E STUN server. The default value is **3478**.

7. Click **Create**. You are returned to the **stun-server** object.
8. Click **Set**. Update and save the configuration.

For more information on the **stun-server** object, see the Oracle Communications OS-E Objects and Properties Reference Guide.

Configuring Encryption

Although the OS-E supports encryption, it does not require it from WebRTC endpoints. If an endpoint does not support encryption, it does not include a crypto key in its answer SDP and RTP is automatically used to transport media.

Because the OS-E always sends media encrypted out, you must configure the in-leg to allow encryption and the out-leg to require it.

You can configure the OS-E to use SDES-SRTP, DTLS, or specify multiple and let the WebRTC endpoint decide which type of encryption to use.

Note: If you configure **encryption-preferences** but do not have **type** set to **multiple**, it does not work. If you specify **multiple** but do not configure **encryption-preferences**, you receive an error.

To configure in-leg encryption:

1. Click the **Configuration** tab and select either **default-session-config** or **session-config-pool > entry**.
2. Click **Configure** next to **in-encryption**.

Configuration
Root path: /

Home Configuration Status Call Logs Event Logs Actions Services Keys Access Tools

Status Summary Logout admin

Configuration: all

Configuration Setup View

- cluster
 - box 1
- vsp
 - default-session-config
 - tls
 - static-stack-settings
 - session-config-pool
 - dial-plan
 - enterprise
 - accounting

Configure vspldefault-session-config/in-encryption Show advanced Help Index

Set Reset Back Delete

mode	allow	(Allow endpoint to offer encryption)
type	RFC-3711	(RFC-3711 compliant SRTP)
require-tls	false	
encryption-preferences	Add encryption-preferences	

3. **mode**—Select **allow** from the drop-down list. This allows the OS-E to receive encryption on the in-leg.
4. **type**—Select the type of encryption you want to use from the drop-down list.
 - RFC3711—Use the SDES-SRTP protocol for encryption.
 - DTLS—Use the DTLS protocol for encryption.
 - **multiple**—Both SDES-SRTP and DTLS are offered for encryption. Using the **encryption-preferences** property, assign each protocol a priority and the type of encryption used depends upon the WebRTC endpoint.
5. If you set **type** to **multiple**, click **Add encryption-preferences** and click **Edit**.

The screenshot shows the Acme Packet Configuration interface. The top navigation bar includes 'Home', 'Configuration', 'Status', 'Call Logs', 'Event Logs', 'Actions', 'Services', 'Keys', 'Access', and 'Tools'. The main content area is titled 'Configure vsp/default-session-config/in-encryption/encryption-preferences'. It features a 'Set' button, a 'Reset' button, a 'Back' button, and a 'Delete' button. Below these buttons are two input fields: 'priority' with a value of '1' and a note '(from 0 to 255)', and 'type' with a dropdown menu set to 'DTLS'. The left sidebar shows a tree view of the configuration hierarchy, including 'cluster', 'box 1', 'vsp', 'default-session-config', 'tls', and 'static-stack-settings'.

6. **priority**—Enter a 1.
7. **type**—Select **DTLS** from the drop-down list.

Note: Always give DTLS a priority of 1 and RFC-3711 a priority of 2.
8. Click **Set**.
9. Click **Add encryption-preferences** and click **Edit**.
10. **priority**—Enter a 2.
11. **type**—Select RFC3711 from the drop-down list.
12. Click **Set**. Update and save the configuration.

To configure out-leg encryption:

1. Click the **Configuration** tab and select either **default-session-config** or **session-config-pool > entry**.

2. Click **Configure** next to **out-encryption**.

The screenshot shows the Acme Packet configuration interface. The top navigation bar includes 'Home', 'Configuration', 'Status', 'Call Logs', 'Event Logs', 'Actions', 'Services', 'Keys', 'Access', and 'Tools'. The main content area is titled 'Configure vsp/default-session-config/out-encryption'. It features a 'mode' dropdown set to 'require', a 'type' dropdown set to 'RFC-3711', and a 'require-tls' dropdown set to 'false'. There is also a link for 'Add encryption-preferences'.

3. **mode**—Select **require** from the drop-down list. This allows the OS-E to offer encryption.
4. **type**—Select the type of encryption you want to use from the drop-down list.
 - RFC3711—Use the SDES-SRTP protocol for encryption.
 - DTLS—Use the DTLS protocol for encryption.
 - multiple—Both SDES-SRTP and DTLS are offered for encryption. Using the **encryption-preferences** property, assign each protocol a priority and the type of encryption used depends upon the WebRTC endpoint.
5. If you set **type** to **multiple**, click **Add encryption-preferences** and click **Edit**.

The screenshot shows the Acme Packet configuration interface for 'encryption-preferences'. The main content area is titled 'Configure vsp/default-session-config/out-encryption/encryption-preferences'. It features a 'priority' input field with the value '2' and a 'type' dropdown set to 'RFC_3711'.

6. **priority**—Enter a **1**.
 7. **type**—Select **DTLS** from the drop-down list.
- Note:** Always give DTLS a priority of 1 and RFC-3711 a priority of 2.

8. Click **Set**.
9. Click **Add encryption-preferences** and click **Edit**.
10. **priority**—Enter a 2.
11. **type**—Select RFC3711 from the drop-down list.
12. Click **Set**. Update and save the configuration.

The **show ice-dtls-status** status provider provides information per call-leg for sessions using DTLS encryption.

```
OS-E>show ice-dtls-status
```

```
session-id: 0x4c40106b423123b
leg: 1
stream: 0
address: 172.30.12.82:24472
remote: 172.30.12.82:24352
type: 1-RTP
role: Passive
state: Succeed
```

Field	Description
session-id	The unique ID of the OS-E session.
leg	Specifies in-leg (0) or out-leg (1).
stream	The media stream index, either audio (0) or video (1).
address	The local OS-E IP and port for this DTLS socket.
remote	The remote peer IP and port for this DTLS socket.
type	Specifies the type of ICE port, either RTP (1) or RTCP (2).
role	Specifies the DTLS role, either Passive or Active.
state	The state of the DTLS socket, either Connected, Listening, Succeeded, or Closed.

Disabling the DTLS Cookie Exchange

For WebRTC to work, you must configure the OS-E to stop exchanging cookies during the DTLS negotiation.

To stop the DTLS cookie exchange:

1. Click the **Configuration** tab and select the **vsp > tls** object.

- Click **Configure** next to **default-dtls-settings**.

The screenshot shows the Acme Packet Configuration interface. The main content area is titled "Configure vsptls/default-dtls-settings". It features a table of configuration parameters:

dynamic-buffers	enabled	(Resource is active)
enable-cbc-counter-measure	true	
tx-record-length	2048	(from 1,024 to 16,384,default=2048)
dynamic-certificate-country-code		
dynamic-certificate-organization-name	DTLS	
dynamic-certificate-common-name	dtls.invalid	
dynamic-certificate-dns-name	dtls.invalid	
dynamic-certificate-days-valid	1	
dtls-cookie-exchange	enabled	(Resource is active)

- dtls-cookie-exchange**—Set to **disabled** to stop exchanging cookies during the DTLS negotiation.
- Click **Set**. Update and save the configuration.

RTP/RTCP Multiplexing

The OS-E supports RTP/RTCP multiplexing. When enabled, the OS-E bundles all of the RTP and RTCP media through the same port.

When initiating a bundled call, the OS-E inserts the necessary information into the INVITE message's SDP in the following format:

```
m=RTP Port
a=rtcp=RTP Port
a=rtcp-mux
```

If the recipient supports RTP/RTCP multiplexing, it returns the following in the SDP of its 200 OK response:

```
m=RTP/RTCP Port
a=rtcp-mux
```

If the recipient does not support RTP/RTCP multiplexing, it returns its own RTP and RTCP port numbers in the SDP without `a=rtcp-mux` and multiplexing is not used.

The OS-E does not support audio and video multiplexing, which is audio and video streams bundled on the same port. To ensure the recipient the OS-E is talking to knows this, you must strip out any Synchronization Source (SSRC) information from the SDP.

To configure RTP/RTCP multiplexing for incoming WebRTC calls:

1. Click the **Configuration** tab and select either **default-session-config** or **session-config-pool > entry**.
2. Click **Configure** next to **in-sdp-attribute-settings**.

The screenshot shows the Acme Packet configuration interface. The top navigation bar includes 'Home', 'Configuration', 'Status', 'Call Logs', 'Event Logs', 'Actions', 'Services', 'Keys', 'Access', and 'Tools'. The main content area is titled 'Configure vsp/default-session-config/in-sdp-attribute-settings'. It features a table with three rows of configuration options:

Attribute	Value	Notes
rtcp-mux	enabled	(Resource is active)
ssrc-in-sdp	pass	
patch-audio-group	enabled	(Resource is active)

3. **rtcp-mux**—Enables or disables RTP/RTCP multiplexing. By default this is **disabled**.
4. **ssrc-in-sdp**—Set to **strip** to strip out any SSRC information from the SDP.
5. **patch-audio-group**—*Advanced property*. Set to **enabled**. When the OS-E receives an offer SDP with both audio and video and the line `a=group BUNDLE audio video` and a response with only audio, it must perform certain functions in order to get the audio to work.

When enabled, the OS-E performs the following modifications:

- The OS-E performs RTP/RTCP multiplexing on the in-leg, regardless of the user configuration
- The OS-E adds bundling information by adding the following to the SDP

```
a=group BUNDLE audio
a=mid:audio
```

- The OS-E generates WebRTC-style SSRC values and adds them to the SDP as well as the RTP/RTCP stream.

Note: To view Advanced properties, you must click the **Show advanced** button.

- Click **Set**. Update and save the configuration.

To configure RTP/RTCP multiplexing for outgoing WebRTC calls:

- Click the **Configuration** tab and select either **default-session-config** or **session-config-pool > entry**.
- Click **Configure** next to **out-sdp-attribute-settings**.

The screenshot shows the Acme Packet Configuration interface. The top navigation bar includes 'Home', 'Configuration', 'Status', 'Call Logs', 'Event Logs', 'Actions', 'Services', 'Keys', 'Access', and 'Tools'. The main content area is titled 'Configure vspdefault-session-config/out-sdp-attribute-settings' and includes a 'Show advanced' button and a 'Help' link. Below the title are 'Set', 'Reset', 'Back', and 'Delete' buttons. The configuration table shows two settings: 'rtcp-mux' set to 'enabled' (with a note '(Resource is active)') and 'ssrc-in-sdp' set to 'pass'. On the left, a tree view shows the configuration hierarchy: cluster > box 1 > vsp > default-session-config > out-sdp-attribute-settings.

- rtcp-mux**—Enables or disables RTP/RTCP multiplexing. By default this is **disabled**.
- ssrc-in-sdp**—Set to **strip** to strip out any SSRC information from the SDP.

Note: To view Advanced properties, you must click the **Show advanced** button.

- Click **Set**. Update and save the configuration.

Configuring SDP Regeneration

To ensure the OS-E represents itself properly in the SDP, it must regenerate incoming SDPs to list the attributes it supports and strip out unsupported attributes. To do this, you must configure the **sdp-regeneration** object.

Note: If the OS-E forwards an SDP containing attributes it does not support, the WebRTC call will not work.

To configure SDP regeneration:

- Click the **Configuration** tab and select either **default-session-config** or **session-config-pool > entry**.

2. Click **Configure** next to **sdp-regeneration**. The **sdp-regeneration** object appears.

The screenshot shows the Acme Packet Configuration interface. The top navigation bar includes 'Home', 'Configuration', 'Status', 'Call Logs', 'Event Logs', 'Actions', 'Services', 'Keys', 'Access', and 'Tools'. The main content area is titled 'Configure vsp/default-session-config/sdp-regeneration' and includes 'Help' and 'Index' links. Below the title are 'Set', 'Reset', 'Back', and 'Delete' buttons. The configuration is presented as a table with various parameters and their current values.

regenerate	enabled	(Resource is active)
origin	rewrite	(Rewrite required SDP parameter to Net-Net OS-E value.)
username		
session-name	rewrite	(Rewrite required SDP parameter to Net-Net OS-E value.)
name		
session-info	strip	(Strip optional SDP parameter.)
uri	strip	(Strip optional SDP parameter.)
e-mail-address	strip	(Strip optional SDP parameter.)
phone-number	strip	(Strip optional SDP parameter.)
bandwidth	strip	(Strip optional SDP parameter.)
timing	pass	(Pass optional SDP parameter unchanged.)
remove-unknown	enabled	(Resource is active)
add-session-connection	disabled	(Resource is inactive)
remove-media-connection	disabled	(Resource is inactive)
add-rtptime	enabled	(Resource is active)
pass-attribute	Edit_pass-attribute	

3. **regenerate**—Set to **enabled** to regenerate the SDP, with the configured settings, before forwarding it along.
4. **add-rtptime**—Set to **enabled** so the OS-E includes rtpmap attributes for well-known CODECs when the rtpmap is not included in the SDP by the original endpoint.

5. `pass-attribute`—Click **Edit pass-attribute**.

The screenshot shows the Acme Packet configuration interface. The top navigation bar includes 'Home', 'Configuration', 'Status', 'Call Logs', 'Event Logs', 'Actions', 'Services', 'Keys', 'Access', and 'Tools'. The main content area is titled 'Configure vspldefault-session-config\sdp-regeneration pass-attribute'. It features a 'Back' button and a list of attributes to be added to the SDP. Each attribute is entered into a text field, followed by 'or select from' and a dropdown menu labeled 'sdplang'. The attributes listed are: ice-ufrag, ice-pwd, candidate, remote-candidates, rtcp, rtcp-mux, and ssrc. Below the list are 'Add', 'Remove All', and 'OK' buttons.

6. Enter the attributes to be included in the SDP. The following attributes must be added:

- ice-ufrag
- ice-pwd
- candidate
- remote-candidates
- rtcp
- rtcp-mux
- ssrc

You must enter attributes one at a time. After entering an attribute and clicking **Add**, a new field to enter the next attribute appears.

Note: These attributes do not appear in the drop-down list and must be entered into the provided blank field.

7. Click **OK**. You are returned to the **sdp-regeneration** object.
8. Click **Set**. Update and save the configuration.

Configuring WebRTC Using SIP Signaling Over WebSockets

One of the ways the OS-E implements signaling for WebRTC is via SIP over WebSockets. WebRTC applications can use JavaScript SIP stack APIs to perform signaling over a websocket and manages STUN requests and WebRTC media.

In addition to UDP, TCP, and TLS transport protocols, the OS-E supports two WebSocket-specific transport protocols. The ws-port is an unencrypted protocol and the wss-port is encrypted with TLS. When using wss-port, a certificate (configured under the **vsp > tls > certificate** object) is required.

SIP transport protocols can be configured as two types of sockets on the OS-E, listener sockets and outgoing connection sockets.

Configuring WebSocket Listener Sockets

To configure WebSocket listener sockets:

1. Click the **Configuration** tab and select the **cluster > box > interface > ip** object.
2. Click **Configure** next to **sip**.
3. Click **Add ws-port**.
4. **port**—Enter the port for the listener socket. There is no default port.
5. Click **Create**. The **ws-port** object appears.

The screenshot shows the Acme Packet Configuration interface. The breadcrumb path is **Configure cluster > box 1 > interface eth0 > ip > sip > ws-port 5000**. The configuration table is as follows:

Field	Value	Notes
port	5000	(from 0 to 65,535)
admin	enabled	(Resource is active)
resource-path	/sip	
http-authentication	type: none	(No HTTP authentication for WebSockets)
http-authentication-realm		

6. **admin**—Specify whether this port is enabled or disabled. The default value is **enabled**.

7. **resource-path**—Specify the HTTP resource to expect in the HTTP GET message. The default value is **/sip**.
8. **http-authentication**—Specify the type of HTTP authentication, if any, that should be applied to the incoming GET message.

Note: No browser currently supports HTTP authentication of a WebSocket, so this property should be left as **none**.

9. **http-authentication-realm**—Specify the realm to use for HTTP authentication when enabled.

Note: Since no browsers currently support HTTP authentication of a WebSocket, this property should be left blank.

10. Click **Set**. Update and save the configuration.

Configuring Secure WebSocket Listener Sockets

When you configure secure WebSocket listener sockets, you must upload a certificate to the **vsp > tls > certificate** object.

To upload a certificate to the OS-E:

1. Click the Configuration tab and select the vsp object.
2. Click **Configure** next to **tls**.

The screenshot shows the acmeApacket Configuration interface. The top navigation bar includes tabs for Home, Configuration, Status, Call Logs, Event Logs, Actions, Services, Keys, Access, and Tools. The main content area is titled "Configuration: all" and shows a tree view on the left with the following structure:

- cluster
 - box 1
- vsp
 - default-session-config
 - tls
 - static-stack-settings
 - session-config-pool
 - dial-plan
 - enterprise
 - accounting

The right pane shows the "Configure vsp:tls" configuration page. It includes buttons for Set, Reset, Back, and Delete. Below these buttons is a table with the following rows:

default-outgoing-settings	Configure
default-ca	Configure
default-crl	Configure
certificate	Add certificate

3. Click **Add certificate**.
4. **name**—Enter a certificate name.

5. Click **Create**.

The screenshot shows the Acme Packet Configuration interface. The top navigation bar includes 'Home', 'Configuration', 'Status', 'Call Logs', 'Event Logs', 'Actions', 'Services', 'Keys', 'Access', and 'Tools'. The 'Configuration' tab is active. The main content area is titled 'Configure vsplit/certificate cert2' and includes a 'Show advanced' button and 'Help' and 'Index' links. Below this are 'Set', 'Reset', 'Back', 'Copy', and 'Delete' buttons. The configuration form has a 'general:' section with three fields: '* name' (containing 'cert2'), 'certificate-file' (with a 'Browse System Files' link), and 'passphrase-tag' (with a 'Manage Password' link). A left sidebar shows a tree view with 'cluster' expanded to 'box 1', and 'vsp' expanded to 'default-session-config', 'tls', 'static-stack-settings', 'session-config-pool', 'dial-plan', and 'enterprise'. The 'Configuration: all' section has 'Configuration', 'Setup', and 'View' buttons.

6. **certificate-file**—Specify the name of the certificate file to use to establish connections with this object. The OS-E supports the following certificate formats:
- PKCS#12—Public Key Cryptography Standard #12 Format (binary)
 - PEM—Privacy Enhanced Mail Format (ASCII)
7. **passphrase-tag**—Specify the passphrase associated with the certificate file. This passphrase must match the string that the certificate was encrypted with.
8. Click **Set**. Update and save the configuration.

To configure secure websocket listener sockets:

1. Click the **Configuration** tab and select the **cluster > box > interface > ip** object.
2. Click **Configure** next to **sip**.
3. Click **Add wss-port**.
4. **port**—Enter the port for the listener socket. There is no default port.

- Click **Create**. The **wss-port** object appears.

The screenshot shows the acme packet configuration interface. The main title is "Configuration" with a "Root path: /" indicator. Below the title is a navigation bar with buttons for Home, Configuration, Status, Call Logs, Event Logs, Actions, Services, Keys, Access, and Tools. The left sidebar shows a tree view of the configuration hierarchy: cluster > box 1 > interface eth0 > ip a > wss-port 5001. The main content area is titled "Configure clusterbox 1interface eth0ip alsiplwss-port 5001" and includes buttons for Set, Reset, Back, Copy, and Delete. The configuration table below is as follows:

* port	5001 (from 0 to 65,535)
admin	enabled (Resource is active)
resource-path	/sip
http-authentication	type: none (No HTTP authentication for WebSockets)
http-authentication-realm	
certificate	Create

- admin**—Specify whether this port is enabled or disabled. The default value is **enabled**.
- resource-path**—Specify the HTTP resource to expect in the HTTP GET message. The default value is **/sip**.
- http-authentication**—Specify the type of HTTP authentication, if any, that should be applied to the incoming GET message.

Note: No browser currently supports HTTP authentication of a WebSocket, so this property should be left as **none**.

- certificate**—Select a certificate for this port from the drop-down list.

Note: This certificate is configured under the **vsp > tls > certificate** object. To configure a new certificate click **Create**. If no certificate is specified, the certificate the OS-E uses for SIP is used.

- Click **Set**. Update and save the configuration.

WebRTC Using OS-E REST Call Control APIs

In WebRTC implementations using OS-E Call Control REST APIs, the web application instructs the OS-E to perform SIP signaling by calling OS-E call control APIs.

This section describes the OS-E APIs and their arguments used in the WebRTC implementation.

Note: Arguments surrounded by angle brackets (< >) are required and arguments surrounded by brackets ([]) are optional.

Register

Executes a WebRTC call using the OS-E's REST APIs. This action allows you to bind a web endpoint to a particular URI. It creates a location cache entry and a unique binding that ties the remote application to the specified URI, allowing remote applications to start receiving calls for the URI without the need to statically configure a dial-plan that routes the calls to a web endpoint.

The URI is a SIP URI in the following formats:

```
sip:user@domain:port
```

When the **register** action is executed, the OS-E first verifies that the user has permission to register that URI. If not, the OS-E returns an “unauthorized” error message.

If the URI is valid, the OS-E performs a registration-plan lookup. If no matches are found, the OS-E returns a “no routes” error message. If a match is found, the OS-E creates a binding that ties the specified URI with the server returned by the registration-plan lookup. Along with the binding, the OS-E also creates an identifier that uniquely identifies the binding. This identifier persists throughout the lifetime of the binding. At the completion of a successful binding, the OS-E returns this identifier along with a “success” message.

When the URI sent in the register action is linked to an existing SIP server, the registration is executed asynchronously. The OS-E returns a “pending” message indicating that the application must monitor for a register event containing the result of the asynchronous registration.

Syntax

```
register <URI> [expiration]
```

Arguments

- <URI>—The URI tied to this binding.
- [expiration]—The expiration time of the binding in seconds. If not specified,

Unregister

Disconnects a WebRTC call using OS-E's REST APIs.

Syntax

```
unregister <URI> <binding-identifier>
```

Arguments

- *<URI>*—The URI tied to this binding.
- *<binding-identifier>*—The identifier that uniquely identifies this binding.

call-control-call

Initiates a call using provided To and From SIP URIs.

You can configure the OS-E to add post-dial digits to a call-control-call action by appending the string **postd=digits** to the user portion of the **To** argument.

Syntax

```
call-control-call <to> <from> [requestId] [originatorFirst]  
[async] [transport] [config] [session-id] [content-type] [body]
```

Arguments

- *<to>*—The destination SIP URI of the session.
- *<from>*—The originating SIP URI of the session.
- [*requestId*]—A unique identifier provided by an external application. This value can be used to identify the call in subsequent events and actions. If a *requestId* is specified, there is a corresponding XML element in the event messages generated for the session.
- [*originatorFirst*]—When **enabled** (the default), the originating party is connected first. When **disabled**, the called party is connected first.
- [*async*] —When **enabled**, causes the OS-E to return a response immediately without waiting for the action to complete. When **disabled**, (the default) the OS-E waits for the action to complete before returning a response.
- [*transport*]—The transport method to use for the call. This can be set to **any**, **TCP**, **UDP**, or **TLS**.

- *[config]*—The **session-config** on the OS-E to use to process a call. Use the full path to the **session-config**. For example:
- `vsp\session-config-pool\entry MyConfig`
- Enclose the value in quotation marks when using the CLI.
- *[session-id]* —The optional session ID for the session.
- *[content-type]*—The content type of the message body of the initial call.
- *[body]*—The message body of the initial call.

call-control-ringing

Rings a destination to indicate an incoming call.

Syntax

```
call-control-ringing <handle> [content-type] [body]
```

Arguments

- *<handle>*—Identifies the leg of a session. Handles are returned as part of the *<info>* element of **call-control** results and can be used to manipulate each leg of a call independently.
- *[content-type]*—Specifies the Content-Type: for the indication.
- *[body]*—Specifies the body for the indication.

call-control-redirect

Redirects an initiated call to a new endpoint, prior to the call being answered. This creates a new call leg and cancels the original one.

Syntax

```
call-control-redirect <handle> <endpoint> [config]
```

Arguments

- *<handle>*—Identifies the leg of a session. Handles are returned as part of the *<info>* element of **call-control** results and can be used to manipulate each leg of a call independently.
- *<endpoint>*—The URI of the call's destination.

What is WebRTC?

- [*config*]—The **session-config** on the OS-E to use to process a call. Use the full path to the **session-config**. For example:

```
vsp\session-config-pool\entry MyConfig
```

Enclose the value in quotation marks when using the CLI.

call-control-accept

Accepts an incoming call from an offering endpoint.

Note: You must specify content-type as application/sdp and body as the SDP for the call.

Syntax

```
call-control-accept <handle> [content-type] [body]
```

Arguments

- <*handle*>—Identifies the leg of a session. Handles are returned as part of the <info> element of **call-control** results and can be used to manipulate each leg of a call independently.
- [*content-type*]—Specifies the Content-Type: for the indication.
- [*body*]—Specifies the body for the indication.

call-control-reject

Rejects an incoming call from an offering endpoint.

Syntax

```
call-control-reject <handle> [response-code] [responseText]
```

Arguments

- <*handle*>—Identifies the leg of a session. Handles are returned as part of the <info> element of **call-control** results and can be used to manipulate each leg of a call independently.
- [*response-code*]—The response code to be used in the response.
- [*responseText*]—The text to be included in the response.

call-control-transfer

Transfers the specified call leg to the specified To SIP URI. The original call leg, referred to by its handle, is disconnected. Handle can be thought of as belonging to the party doing the transfer, even though the transfer is done via a third-party action.

Syntax

```
call-control-transfer <handle> <to>
```

Arguments

- *<handle>*—Identifies the leg of a session. Handles are returned as part of the *<info>* element of **call-control** results and can be used to manipulate each leg of a call independently.
- *<to>*—The destination SIP URI of the session.

call-control-insert-dtmf

Inserts DTMF digits into SIP sessions. DTMF is inserted only into the call leg specified; the other party does not hear it.

Syntax

```
call-control-insert-dtmf <handle> <digits> [volume] [duration]
```

Arguments

- *<handle>*—Identifies the leg of a session. Handles are returned as part of the *<info>* element of **call-control** results and can be used to manipulate each leg of a call independently.
- *<digits>*—Specifies the digits inserted into the call leg.
- [*volume*]—The volume of the DTMF digits, in decimals from -36 to 0. The value **1** is the default.
- [*duration*]—The duration of each digit in milliseconds, from 100 to 10000. The value **0** is the default.

call-control-park

Creates a call to an endpoint from a given SIP URI. If you specify a From URI, it is used as the From URI in the SIP message; if you specify no From URI, the From URI is that of the given endpoint.

Syntax

```
call-control-park <endpoint> [from] [requestId] [async] [sessionID]  
[persist] [config]
```

Arguments

- *<endpoint>*—The URI of the call's destination.
- *[from]*—The originating SIP URI of the call.
- *[requestId]*—A unique identifier provided by an external application. This value can be used to identify the call in subsequent events and actions. If a *requestId* is specified, there is a corresponding XML element in the event messages generated for the session.
- *[async]*—When **enabled**, causes the OS-E to return a response immediately without waiting for the action to complete. When **disabled** (the default), the OS-E waits for the action to complete before returning a response.
- *[sessionID]*—The optional session ID for a rendezvous session.
- *[persist]*—When **enabled**, a connected session remains parked even when the remote endpoint disconnects.
- *[config]*—The **session-config** on the OS-E to use to process a call. Use the full path to the **session-config**. For example:

```
vsp\session-config-pool\entry MyConfig
```

Enclose the value in quotation marks when using the CLI.

call-control-attach

Attaches a call leg to an existing SIP session for three-way conferencing.

Syntax

```
call-control-attach <handle> <session-id>
```

Arguments

- *<handle>*—The handle of the endpoint to be attached.
- *<session-id>*—The session to which the endpoint is being attached.

call-control-disconnect

Terminates all parties in a SIP session.

Syntax

```
call-control-disconnect <handle>
```

Arguments

- *<handle>*—Identifies the leg of a session. Handles are returned as part of the *<info>* element of **call-control** results and can be used to manipulate each leg of a call independently.

Chapter 14. Configuring the Multimedia Streaming Server

About This Chapter

This chapter provides information on configuring the multimedia streaming server (MSS), a process enhancing media and signaling capabilities of the OS-E to web and mobile-based applications.

The MSS allows existing SIP and H.323 endpoints to connect to web-based multimedia applications. It also enables a user to communicate to and from SIP/H.323 endpoints without needing a traditional phone. These web-based applications can utilize the microphone, speaker, and webcam from a host computer or mobile device for encoding and decoding media.

The MSS can be configured to connect either externally, with a third-party flash media server to a Flash Video Content Distribution Network (CDN) or directly on the OS-E as an internal media server.

Configuring MSS

This section describes the process to configure MSS on the OS-E.

Configuring an External Flash Media Server

To configure an external flash media server:

1. Select the **Configuration** tab and click the **vsp** object.
2. Click **Configure** next to **multimedia-streaming-config**.

3. Click **Add server**.

The screenshot shows the Acme Packet Configuration web interface. The top navigation bar includes 'Home', 'Configuration', 'Status', 'Call Logs', 'Event Logs', 'Actions', 'Services', 'Keys', 'Access', and 'Tools'. The main content area is titled 'Create vsplmultimedia-streaming-configserver - Step 1 of 1: Edit server'. Below the title, there is a prompt: 'Please provide some basic information for server. Then press "Create".' The form contains three input fields: '* name' with the value 'FMS1', '* host' with the value '10.1.5.11' (with a note '(host name or n.n.n.n)'), and '* port' with the value '1935' (with a note '(from 0 to 65,535)'). At the bottom of the form are three buttons: 'Create', 'Reset', and 'Cancel'. On the left side, there is a tree view showing the configuration hierarchy: 'cluster' > 'box 1' > 'vsp' > 'default-session-config', 'tls', 'static-stack-settings', 'session-config-pool', 'enterprise', 'accounting', 'location-service', 'h323-settings', and 'multimedia-streaming-config'.

4. **name**—Enter a unique name for the flash media server.
5. **host**—Enter the host name or IP address of the flash media server.
6. **port**—Enter the port number for the FMS to listen. The default is **1935**. The minimum is 1024 and the maximum is 65535.
7. Click **Create**.
8. Click **Set**. Update and save the configuration.

When you configure an internal media server, it can be one of three protocols: Real Time Media Protocol (RTMP), Real Time Media Protocol Tunneled (RTMPT), which is RTP tunneled over HTTP, or Real Time Media Protocol Secure (RTMPS), which is RTP over TLS.

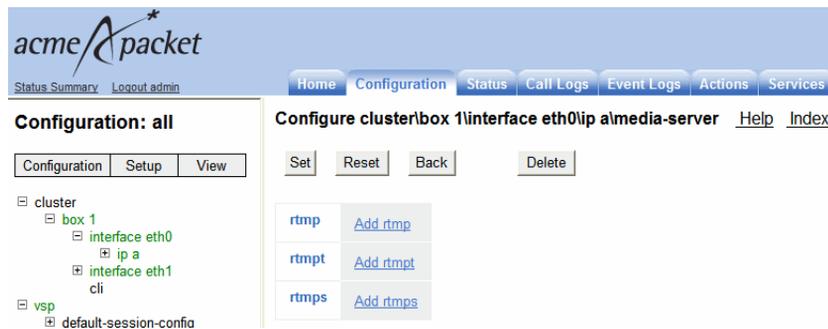
The **media-server** object configures an internal media server. You must configure this server both here and under the **multimedia-streaming-config > server**. You must configure both objects with the same name, IP address, and port.

Configuring an Internal Media Server

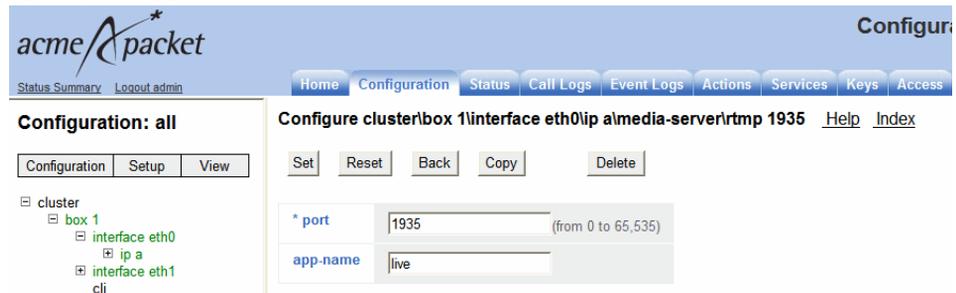
To configure an internal media server:

1. Select the **Configuration** tab and click the **cluster > box > interface > ip** object.
2. Click **Configure** next to **media-server**.

3. Click either **Add rtmp**, **Add rtmpt**, or **Add rtmps** depending on which protocol you want to use for this server.



4. Specify a TCP **port** to use for receiving requests.
5. Click **Create**.
6. Specify a unique **app-name** for this server.



7. Click **Set**.
8. Click the **vsp** object.
9. Click **Configure** next to **multimedia-streaming-config**.
10. Click **Add server**.
11. Specify the same **name** as you did in the **media-server** object.
12. Specify the same **port** as you did in the **media-server** object.
13. Click **Set**. Update and save the configuration.

Types of Supported MSS Calls

The MSS process supports two types of calls, MSS calls and SIP/H.323 audio/video broadcasts.

MSS calls must be initiated using the **call-control call** action via either the REST or SOAP interface. For more information on the **call-control** action, see the Oracle Communications ASC Web Services SOAP/REST API Guide.

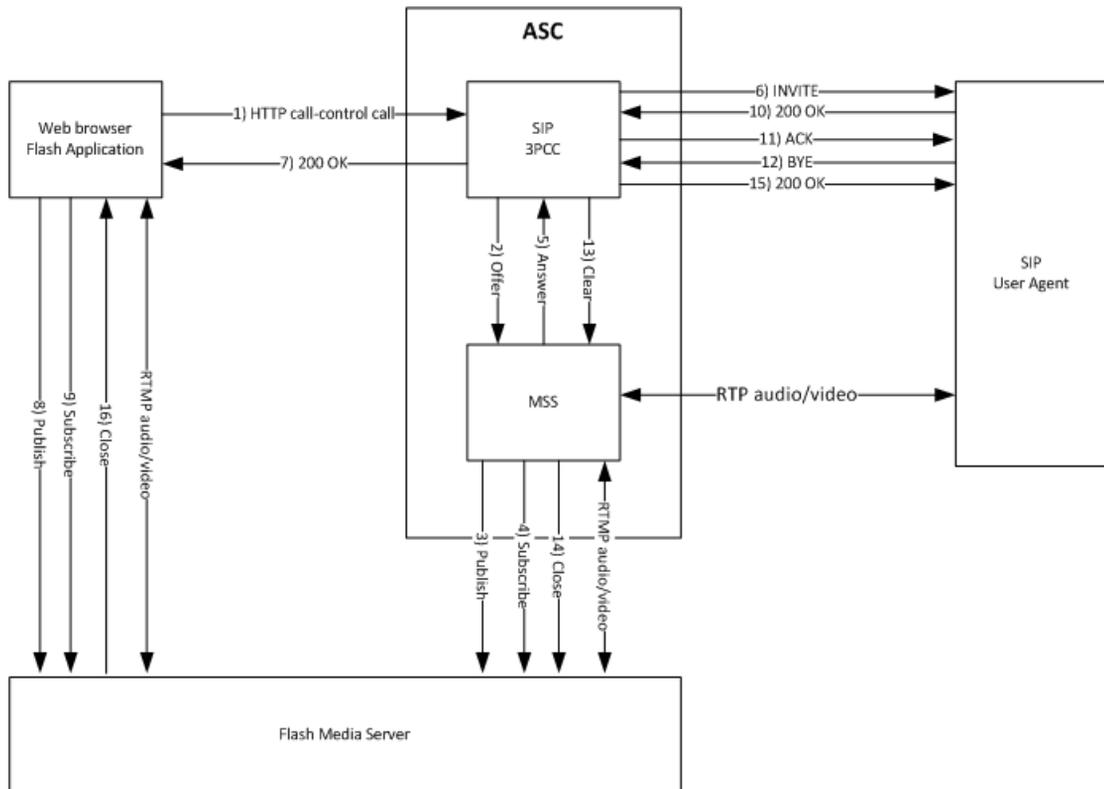
By initiating the calls this way, these call legs can use any of the **call-control** features. The **call-control call from** argument must be identified as `rtmp:user@host` and the **to** argument must be a SIP or H.323 URL. An object, `MediaStreamInfo`, is returned in the call-control call result (when placed synchronously) and in the `CallConnectedEvent` (when placed asynchronously).

The RTP audio from the SIP side of these calls is transcoded from any supported media to Speex, G.711 mu-law, or G.711 a-law on the RTMP side and vice-versa. Video, however, is not transcoded. The OS-E repackages the RFC 3984 encoded H.264 from the RTP side into the Advanced Video Codec (AVC) on the RTMP side and vice versa.

The following is an example call flow of an MSS call along with a corresponding diagram.

1. The Flash application initiates a REST or SOAP ‘call-control call `rtmp:123@foo.com sip:456@bar.com`’.
2. 3PCC performs a dial-plan lookup on `rtmp:123@foo.com` and matches a multimedia-streaming-server. It then sends an offer event to MSS with an SDP containing all supported codecs, as well as the newly created session-ID.
3. MSS sends a PUBLISH to the flash media server for the SIP leg.
4. MSS sends a SUBSCRIBE to the flash media server for the flash leg.
5. MSS removes unsupported CODECS (currently only Speex, G.711 mu-law, G.711 a-law, and H.264 are supported) and sends an answer event to SIP 3PCC.
6. 3PCC sends an INVITE to the SIP user agent.
7. 3PCC sends a 200 OK to the Flash application with the RTMP stream information in the 200 OK (if synchronous) or in the `CallConnected` event (if asynchronous) in the response.

8. The Flash application sends a PUBLISH to the flash media server with the stream name for the Flash leg.
9. The Flash application sends a SUBSCRIBE to the flash media server with the stream name of the SIP leg.
10. The SIP UA sends a 200 OK (with optional 100, 180 before that).
11. 3PCC sends an ACK. Audio and video is now flowing in both directions.
12. The SIP UA sends a BYE.
13. 3PCC sends a clear event to MSS.
14. MSS sends an RTMP stream close, clearing up the stream on the FMS.
15. 3PCC sends a 200 OK to the BYE.
16. FMS sends an RTMP stream close to the Flash application, telling the application that the call has been terminated.



A SIP/H.323 audio/video broadcast is initiated when the OS-E receives an INVITE or SETUP processed by the 3PCC stack. It executes a normal dial-plan lookup where it matches a dial-plan with a peer that references a server type called streamer. The streamer server type lets 3PCC know that it needs to establish a dialog with MSS.

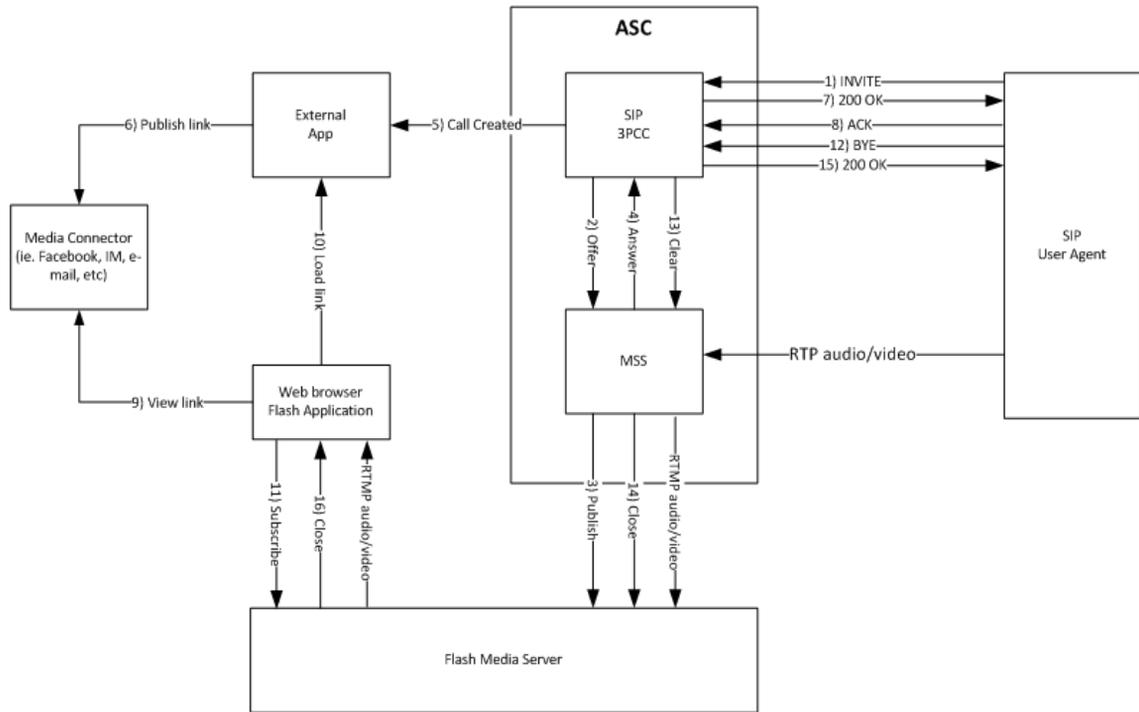
The MSS process then starts sending RTMP media by publishing its audio to the flash media server. Conversely, the MSS starts receiving the RTMP media from the flash endpoint by subscribing to the flash media server with the stream name returned in the MediaStreamInfo object of the CallConnected event.

As with MSS calls, the RTP audio from the SIP side is transcoded from any supported media to Speex on the RTMP. Video, however, is not transcoded. The OS-E repackages the RFC 3984 encoded H.264 from the RTP side into the AVC on the RTMP side.

The following is an example call flow of a SIP/H.323 audio/video broadcast along with a corresponding diagram.

1. The SIP/H.323 endpoint initiates a call to the OS-E.
2. 3PCC performs a dial-plan lookup on the request-URI and matches a multimedia-streaming-server. It then sends an offer event to MSS with an SDP containing all supported codecs, as well as the newly created session-ID.
3. MSS sends a PUBLISH to the flash media server for the SIP leg.
4. MSS removes unsupported CODECS (currently only Speex, G.711 mu-law, G.711 a-law, and H.264 are supported) and sends an answer event to the SIP 3PCC.
5. The 3PCC sends a callCreated event to the external application via external-services events.
6. The external application publishes a link to the live media broadcast via a media connector service such as Facebook, IM or e-mail.
7. The 3PCC sends a 200 OK to the SIP UA.
8. The SIP UA sends an ACK. The audio and video is now broadcast to the web.

- The user browses to the social media connector and sees the link to the live broadcast.



As with all processes on the OS-E, the OS-E creates events for the MSS. For a list and description of all MSS events, see Events in the web services home page's REST documentation.

In order for the OS-E to route calls to the appropriate web endpoint, use the **dial-plan > route > peer type** property option. Use the **streamer** peer type to reference the configured **vsp > multimedia-streaming-config**. You must also have **third-party-call-control** enabled to route this server.

To configure the OS-E to route calls to the appropriate endpoint:

- Select the **Configuration** tab and click the **vsp > default-session-config** or **vsp > session-config-pool > entry** object.
- Click **Configure** next to **third-party-call-control**.
- Set **admin** to **enabled**.
- Click **Set**.

5. Click the **vsp** object.
6. Click **Configure** next to **dial-plan**.
7. Click **Add route** next to **route**.
8. Specify a **name** for this route. Click **Create**.
9. From the **peer type** drop-down box, select **streamer**.
10. From the **streamer** drop-down box, select the **multimedia-streaming-config > server** you want to associate with this **dial-plan**. This is the server you configured to set up either the FMS or the internal media server.
11. Click **Set**. Update and save the configuration.

The action, **stream**, tests the connectivity of a newly configured **multimedia-streaming-server**.

The **stream publish** action allows you to publish an audio or video stream to a specified server. The action syntax is:

```
stream publish [audio-file] [video-file] [server] [stream-name]
```

Valid parameters for this action are:

- *[audio-file]*—The Speex encoded audio file to transmit.
- *[video-file]*—The H.264 encoded video file to transmit.
- *[server]*—The name of the **vsp > multimedia-streaming-config > server** to publish to for this test.
- *[stream-name]*—The name of the stream to which you want to publish.

The **stream subscribe** action allows you to subscribe to an audio or video stream to a specified server. The action syntax is:

```
stream subscribe [audio-file] [video-file] [server] [stream-name]
```

Valid parameters for this action are:

- *[audio-file]*—The file name to use when writing out received audio data.
- *[video-file]*—The file name to use when writing out received video data.
- *[server]*—The name of the **vsp > multimedia-streaming-config > server** to subscribe to for this test.
- *[stream-name]*—The name of the stream from which you want to receive.

Three status providers provide information about the MSS process.

The **show multimedia-streaming-server** action provides information about the multimedia streaming servers configured on the OS-E.

```
NNOS-E>show multimedia-streaming-server
```

```
name                protocol host                port  hits
-----            -
internall1          RTMP    156.40.1.11         1935  0
```

Field	Description
name	The name of the multimedia streaming server.
protocol	The protocol over which this port is listening.
host	The IP address of this server.
port	The port over which this server is listening.
hits	The number of requests currently sent to this server.

The **show multimedia-streaming-peers** action lets you know if your multimedia-streaming servers are configured properly.

The **show multimedia-streaming-pool** action provides information about the configured multimedia streaming pool that has been derived from sip-server-pool.

```
NNOS-E>show multimedia-streaming-pool
```

```
peer-name: internall1
server: internall1
  host: 156.40.1.11
    TPT: any
port: 1935
  box: local
state: up
  in: 0
  out: 0
```

Field	Description
peer-name	The name of this pool's peer.
server	The server associated with this pool.
host	The IP address of this server.

Field	Description
TPT	The preferred transport method of this server. Currently, this is always set to any.
port	The port over which this server is listening.
box	The OS-E where this server is configured. This value is currently always the local OS-E.
state	Whether the server is available (up) or not (down).
in	The number of packets received from this server. This value is currently not applicable.
our	The number of packets sent to this server. This value is currently not applicable.

Appendix A. Interpreting MOS Statistics

About This Appendix

This appendix provides information that will assist you in evaluating Mean Opinion Score (MOS) measurements that appear in the OS-E Management System Call Logs.

Mean Opinion Score Overview

Mean Opinion Score (MOS) is a subjective measurement and an “opinion” of the audio quality heard by the listener on a phone. The MOS measurement reveals the call quality as:

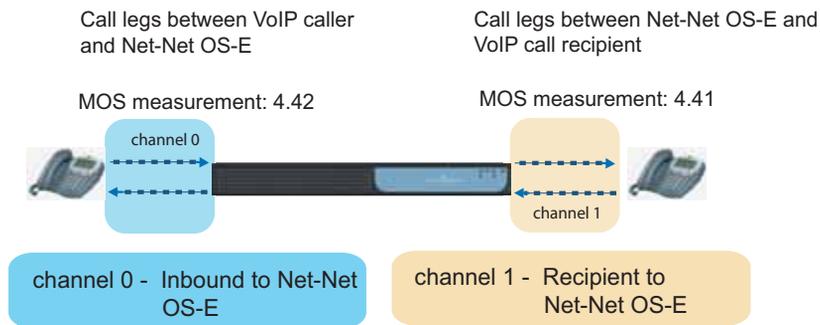
- 1 (bad — pure noise)
- 2 (poor)
- 3 (fair)
- 4 (good)
- 5 (excellent — perfect fidelity)

"Toll quality" sound is generally associated with a MOS score of at least 4. G.711 starts with an intrinsic MOS value of 4.4, while G.729, which performs significant compression, has an intrinsic MOS value of 4.1. The MOS score on a VoIP network is further reduced when there is packet loss, excessive delays, etc.

MOS Call Quality Statistics Gathering

MOS results are derived using a formula that evaluates both the inbound and the outbound VoIP call session, with each call session having a resulting score assigned to it. The inbound call session is the two-channel VoIP media stream between the call originator and the OS-E system; the outbound call session is the two-channel VoIP media stream between the OS-E and VoIP call destination.

The following image illustrates the points in the network where the OS-E takes MOS measurements.



Formulating MOS Results

The OS-E uses a modified MOS E-Model standard algorithm (ETSI technical report ETR 250, G.107 standard) that adds a measurement for call jitter to derive the final MOS calculation. The components applied by this algorithm are:

- **Jitter** (current jitter)—in milliseconds, 0 to 500
- **Delay** (latency)—in milliseconds, 0 to 500
- **Packet loss**—in percent, 0 to 20
- **CODEC**—G.711, G.729, G.723.1, etc. participating in the call legs

It is important to note that the type of CODEC used in the calculation affects the MOS result. This is due to differences in compression and bit rates. See the table below for sample CODEC comparison information.

R-factor Formula

The R-factor is a rating on the overall conversational quality of a call, derived from the E-Model; expressed on a 0-to-100 scale, where 0 is extremely bad quality, and 100 is very high quality. The R-factor result is used in the MOS calculation.

The R-factor formula is:

$$\mathbf{R = R_o - I_s - I_d - I_e}$$

where,

- “**R**” is a recency factor, usually 94 (default).
- “**R_o**” is a base factor determined from noise levels.
- “**I_s**” represents impairments occurring simultaneously with speech.
- “**I_d**” represents end-to-end impairments that are delayed with respect to speech.
- “**I_e**” represents the distortion impairments associated with equipment.

If one segment of the call leg has an R-factor of 69, and if the other segment has an R-factor of 65, the equation would be as follows:

$$\mathbf{R = 94 - (94 - 69) - (94 - 65) = 40}$$

MOS Formula

The MOS result is perceived quality rating obtained from the E-Model R factor on the conversational quality of a call; expressed on a 1-to-5 scale, where 1 is extremely bad quality and 5 is very high quality.

The MOS formula is:

$$\mathbf{MOS = 1 + 0.035R + 7 \times 10^{-6}R(R-60)(100-R)}$$

With an R-factor of 40 applied to *R* in this equation, the resulting MOS score would be 2.06.

For wideband CODECs, the MOS range is 1-5, even though the R-factor range is higher. This means that a narrowband CODEC may have a MOS score of 4.3 and a wideband CODEC may have a MOS score of 3.9, even though the wideband CODEC sounds much better. The following table provides some of the CODEC comparisons.

CODEC	Bit rate (Kbps)	Compression	MOS
G.711	64	2:1	4.1
G723.1	5.3	8:1	3.65
	6.3	7:1	3.9
G.729A	8	8:1	3.7

Displaying MOS Results With the Net-Net OS-E Management System

You can display MOS results in the following ways from the Call Logs function:

1. Select **Sessions**, followed by **Statistics**—To display MOS results and related statistics
2. Select **Accounting Calls**—To display final MOS calculations per call channel
3. Select **Call Record** from the **Sessions** or **Accounting Calls** function—To display final MOS calculations per call channel.



Note: For MOS statistics gathering on the OS-E, make sure that you first enable the **rtp-stats** property in the **media/session-config** object.

The following image illustrates a sample Call Logs Statistics page with the MOS calculation field on the left.

Call Quality Statistics (QoS) for Session 0x04C261EEA043C661

Media Type	uri	mos	timestamp	self-made	begin-ex-seq	end-ex-seq	packets-lost	current-jitter	average-latency	packets-duplicate
audio/pcmu	From sip.richardseymour@nfl.com less...		10:59:09.455 Wed 2006-10-25	true	0	0	0	450	0	0
<ul style="list-style-type: none"> session-id:0x04C261EEA043C661 fraction-lost:0 min-jitter:4 max-jitter:450 min-TTL:64 max-TTL:64 streamindex:0 channel:0 min-latency:0 max-latency:0 scpName: codecs:pcmu telephone-event 		4.41								
audio/pcmu	From sip.tombrady@nfl.com less...		10:59:09.455 Wed 2006-10-25	true	0	0	0	106	0	0
<ul style="list-style-type: none"> session-id:0x04C261EEA043C661 fraction-lost:0 min-jitter:4 max-jitter:120 min-TTL:64 max-TTL:64 streamindex:0 channel:1 min-latency:0 max-latency:0 scpName: codecs:pcmu pcmu g726-32 gsm g729 g723 telephone-event 		4.42								

The following image illustrates a sample Call Logs Accounting Calls page with the per-channel MOS calculations to the far right.

Boston

Search Criteria

Search Type: All Calls

The following image illustrates a Call Record page containing the final formatted MOS calculation.

Boston

Search Criteria

Search Type: All Calls

mimetype-on-dest-leg	audio/pcmu
latency-on-dest-leg	0
rfactor-on-dest-leg	0
rfactor-on-dest-leg	93199
rfactor-on-src-leg	93199
mos-fmt-on-dest-leg	
mos-fmt-on-src-leg	
MOS-on-dest-leg	0
MOS-on-src-leg	0

The following fields, in conjunction with the CODEC information, are significant to the MOS calculation. The latency values are measured only for loopback calls. Otherwise, the latency value displays as 0.

- **packets-lost**—The number of packets dropped during the VoIP call.
- **current-jitter**—The final jitter value for the VoIP call.
- **min-jitter**—The minimum jitter recorded during the duration of the call.
- **max-jitter**—The maximum jitter recorded during the duration of the call.
- **average-latency**—The average latency period, in milliseconds, over the course of the call.
- **min-latency**—The shortest latency experienced, in milliseconds, over the course of the call.
- **max-latency**—The longest latency experienced, in milliseconds, over the course of the call.
- **mos**—The resulting MOS calculation.
- **packets-dropped**—The number of packets received and dropped by the OS-E (including drops while the OS-E is inserting announcements, crypto-related drops, verification drops, and routing error drops).
- **packets-lost**—The missing sequence numbers on the inbound stream (when rtp-stats are turned on).

Appendix B. Call Detail Record Entries

About This Appendix

This appendix provides information on elements that make up the call detail record (CDR) entries that appear in the OS-E call logs.

Call Detail Record Overview

A call detail record (CDR) provides important accounting information about the SIP phone calls (or sessions) that have been processed by the OS-E. The call accounting software retrieves and processes CDR data.

CDRs can be used by a variety of applications to determine information, such as:

- Identities of call sources and call destinations
- Call duration and whether the call was recorded
- Usage and billing information based on connect and disconnect times
- Routing information associated with the call session, such as the SIP URIs and the previous and next hops in the call path
- Mean Opinion Scores (MOS) to determine the quality of the call, as covered in Appendix A, “Interpreting MOS Statistics.”



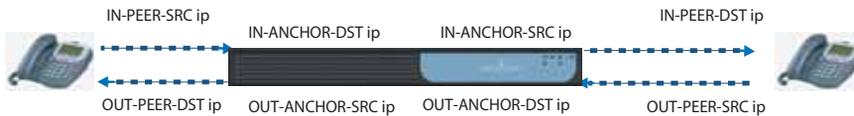
Note: recommends that you use the OS-E Management System **Call Logs** function to view call detail records. Viewing call detail records outside of the OS-E Management System will not reveal the column headers that identify the individual CDR fields.

SIP Call Legs

The following image shows both the inbound and the outbound VoIP call sessions. The inbound call session on channel 0 represents VoIP media stream between the call originator, the OS-E system, and the call destination. The outbound call session on channel 1 represents the VoIP media stream between the call destination, the OS-E system, and back to the call originator (or the source). The OS-E Management System **Sessions** page shows the session information associated with the call.

Call legs between VoIP source phone and Net-net OS-E

Call legs between Net-Net OS-E and VoIP destination phone



The screenshot shows the 'Call Logs' page in the acme packet management system. The page includes a navigation menu with options like Home, Configuration, Status, Call Logs, Event Logs, Actions, Services, Keys, Access, Tools, and Portal. A search box is present with 'Search Type' set to 'All Sessions' and a 'Search' button. Below the search box, the page indicates 'Page 1 of 1 showing 30 items'. A table of sessions is displayed with columns for Created, Method, Result, From, To, Call ID, Session ID, and Type.

Created	Method	Result	From	To	Call ID	Session ID	Type
15:21:09:324 Thu 2008-10-16	INVITE	Code2xx	sip:5551212.com	sip:5551212.com	CXC-4-5a7ab20-99001aac-13c4-48079424-3a54a8-3a977b1@5551212.com	0a04c312c30c7DE772	B2BUA
15:20:07:455 Thu 2008-10-16	INVITE	Code2xx	sip:172.30.1.42	sip:172.30.1.42	CXC-3-5a7ab130-99001aac-13c4-480793a6-3c7310-23595b6@172.30.1.42	0a04c312c300FF1997	B2BUA

The following image shows the Call Logs\Statistics page. Note that the call session shows the duplicate timestamp for each call leg, indicating the time when the call ends and when the OS-E writes the call record to call log database. Since both call legs end at the same time, they have the same timestamp.

The screenshot displays the 'Call Quality Statistics (QoS) for Session 0x04C261EEA043C661'. It features a table with the following columns: Media Type, uri, mos, timestamp, self-made, begin-ex-seq, end-ex-seq, packets-lost, current-jitter, average-latency, and packets-duplicate. Two call legs are listed, both with a timestamp of 10:59:09.455 Wed 2006-10-25. The first leg has 450 packets-lost and 0 jitter, while the second leg has 106 packets-lost and 0 jitter. Both legs have a session-id of 0x04C261EEA043C661. Below the table, there are two expandable sections showing detailed statistics for each leg, including session-id, fraction-lost, min-jitter, max-jitter, min-TTL, max-TTL, streamindex, channel, min-latency, max-latency, scpName, and codecs.

Media Type	uri	mos	timestamp	self-made	begin-ex-seq	end-ex-seq	packets-lost	current-jitter	average-latency	packets-duplicate
audio/pcmu	From sip.richardseymour@nfl.com less...		10:59:09.455 Wed 2006-10-25	true	0	0	0	450	0	0
<ul style="list-style-type: none"> session-id:0x04C261EEA043C661 fraction-lost:0 min-jitter:4 max-jitter:450 min-TTL:64 max-TTL:64 streamindex:0 channel:0 min-latency:0 max-latency:0 scpName: codecs:pcmu telephone-event 										
audio/pcmu	From sip.tombrady@nfl.com less...		10:59:09.455 Wed 2006-10-25	true	0	0	0	106	0	0
<ul style="list-style-type: none"> session-id:0x04C261EEA043C661 fraction-lost:0 min-jitter:4 max-jitter:120 min-TTL:64 max-TTL:64 streamindex:0 channel:1 min-latency:0 max-latency:0 scpName: codecs:pcmu pcm a g726-32 gsm g729 g723 telephone-event 										

Displaying the CDR

You can display CDRs in the following ways from the **Call Logs** function:

1. Select **Sessions**, followed by **Call Record**.
2. Select **Accounting Calls**, followed by **Call Record**.



Note: For MOS statistics gathering on the OS-E system, make sure that you first enable the **rtp-stats** property in the **media/session-config** object.

The following image illustrates an Accounting Call Record page in the OS-E Management System.

The screenshot displays the 'Accounting Call Record' page in the OS-E Management System. The page includes a navigation menu at the top with options like Home, Configuration, Status, Call Logs, Event Logs, Actions, Services, Keys, Access, Tools, and Portal. On the left, there is a 'Select:' menu with options for Sessions, User Sessions, SIP Messages, Queues, Accounting Calls, and Files. The main content area shows a detailed view of a call record with the following fields and values:

session-id	Dx04C261EEA043C661
recorded	true
call-id	3c40de9fc0df-ve3zh6o1nwtg@enom360
to	<sip.richardseymour@nfl.com>
from	"tombrady" <sip.tombrady@nfl.com>;tag=ukw6zopyih
method	INVITE
incoming-request-uri	sip.richardseymour@nfl.com
previous-hop-ip	192.168.215.12
previous-hop-via	SIP/2.0/UDP 192.168.215.12.2051;branch=z9hG4bK-0stgkyd4atr;port
outgoing-request-uri	sip.richardseymour@192.168.215.11.2051;time=zpbidz#
next-hop-ip	192.168.215.11
next-hop-dn	192.168.215.11
header	3c40de9fc0df-ve3zh6o1nwtg@enom360
origin	
setup-time	10:59:00.245 Wed 2006-10-25
connect-time	10:59:02.165 Wed 2006-10-25
disconnect-time	10:59:09.451 Wed 2006-10-25
disconnect-cause	Bye
scp-name	default
call-id-2	
packets-received-on-src-leg	357
packets-lost-on-src-leg	0
packets-discarded-on-src-leg	3
pdv-on-src-leg	450
codec-on-src-leg	pcmu telephone-event
mimetype-on-src-leg	audio/pcmu
latency-on-src-leg	0
rfactor-on-src-leg	0
packets-received-on-dest-leg	364
packets-lost-on-dest-leg	0
packets-discarded-on-dest-leg	4

CDR Field Descriptions and Data Types

The following table lists and describes the fields and data types that make up a call detail record.

CDR field	MS-SQL data types	PostgreSQL data types	Oracle data types	Description
session-id	type="uint64" format="hex" key="index"	type="int8"	NUMBER	The unique session identifier in hexadecimal format, unassigned 64-bit integer.
recorded	type="Boolean"	type="int4"	NUMBER	The true or false indication as to whether the SIP call was recorded.
call-id	type="String"	type="name"	VARCHAR2 (256)	The unique call identifier of the inbound call leg.
to	type="String"	type="name"	VARCHAR2 (256)	The string in the To URI: field of the SIP header.
from	type="String"	type="name"	VARCHAR2 (256)	The string in the From URI:field of SIP header.
method	type="String"	type="name"	VARCHAR2 (256)	The SIP method, such as INVITE or REGISTER, that initiated the call session.
incoming-request-uri	type="String"	type="name"	VARCHAR2 (256)	The Request URI on the inbound call leg.
previous-hop-ip	type="IPHost"	type="int4"	NUMBER	The IP address of the previous hop; the last network node handling the call before received at the OS-E system.
previous-hop-via	type="String"	type="name"	VARCHAR2 (256)	The VIA header from the previous hop.
outgoing-request-uri	type="String"	type="name"	VARCHAR2 (256)	The Request URI on the outbound leg.
next-hop-ip	type="IPHost"	type="int4"	NUMBER	The IP address of the next hop; the next network node handling the call forwarded by the OS-E.
next-hop-dn	type="String"	type="name"	VARCHAR2 (256)	The fully qualified domain name (FQDN) or IP address of the next network node handling the call forwarded by the OS-E.

CDR field	MS-SQL data types	PostgreSQL data types	Oracle data types	Description
header	type="String"	type="name"	VARCHAR2 (256)	An arbitrary header associated with the SIP call.
origin	type="String"	type="name"	VARCHAR2 (256)	The ORIGIN header associated with the SIP call.
setup-time	type="Time" key="index"	type="timestamp"	TIMESTAMP	The time at which the SIP was set up at the OS-E in the format <i>hour:minutes:seconds.millisecond s: weekday year-month-day</i> .
connect-time	type="Time"	type="timestamp"	TIMESTAMP	The time at which the SIP was connected to the SIP call destination in the format <i>hour:minutes:seconds.millisecond s: weekday year-month-day</i> .
disconnect-time	type="Time"	type="timestamp"	TIMESTAMP	The time at which the SIP was disconnected from the SIP call destination in the format <i>hour:minutes:seconds.millisecond s: weekday year-month-day</i> .
disconnect-cause	type="DisconnectType"	type="int4"	NUMBER	The reason for the call disconnection, such as BYE.
duration	type="uint32"	type="int4"	NUMBER	Duration of the call in seconds.
scp-name	type="String"	type="name"	VARCHAR2 (256)	The OS-E virtual system partition (VSP) that handled the call.
call-id-2	type="String"	type="name"	VARCHAR2 (256)	The secondary call identifier for the outgoing leg.
origGW	type="String"	type="name"	VARCHAR2 (256)	The name of the originating gateway associated with the call.
termGW	type="String"	type="name"	VARCHAR2 (256)	The name of the gateway where the call was terminated.
packets-received-on-src-leg	type="uint32"	type="int4"	NUMBER	The total number of packets received on the inbound call leg.
packets-lost-on-src-leg	type="uint32"	type="int4"	NUMBER	The total number of packets lost on the inbound call leg.
packets-discarded-on-src-leg	type="uint32"	type="int4"	NUMBER	The total number of packets discarded on the inbound call leg.

CDR field	MS-SQL data types	PostgreSQL data types	Oracle data types	Description
pvd-on-src-leg	type="uint32"	type="int4"	NUMBER	The packet delay variation (jitter) associated with the call on the inbound call leg.
codec-on-src-leg	type="String"	type="name"	VARCHAR2 (256)	The CODEC associated with the inbound call leg.
mimetype-on-src-leg	type="String"	type="name"	VARCHAR2 (256)	The MIME type associated with the inbound call leg, such as audio/pcmu.
latency-on-src-leg	type="uint32"	type="int4"	NUMBER	The total processing time of the inbound call leg.
rfactor-on-src-leg	type="uint16" or type="uint32"	type="int4"	NUMBER	The R-factor integer used in the MOS score compilation on the inbound call leg.
packets-received-on-dest-leg	type="uint32"	type="int4"	NUMBER	The total number of packets received on the outbound call leg.
packets-lost-on-dest-leg	type="uint32"	type="int4"	NUMBER	The total number of packets lost on the outbound call leg.
packets-discarded-on-dest-leg	type="uint32"	type="int4"	NUMBER	The total number of packets discarded on the outbound call leg.
pvd-on-dest-leg	type="uint32"	type="int4"	NUMBER	The packet delay variation (jitter) associated with the call on the outbound call leg.
codec-on-dest-leg	type="String"	type="name"	VARCHAR2 (256)	The CODEC associated with the outbound call leg.
mimetype-on-dest-leg	type="String"	type="name"	VARCHAR2 (256)	The MIME type associated with the outbound call leg, such as audio/pcmu.
latency-on-dest-leg	type="uint32"	type="int4"	NUMBER	The total processing time of the outbound call leg.
rfactor-on-dest-leg	type="uint16"	type="int4"	NUMBER	The R-factor integer used in the MOS score compilation on the destination call leg.

CDR field	MS-SQL data types	PostgreSQL data types	Oracle data types	Description
Rx1000FactorOnDestLeg	type="uint32"	type="int4"	NUMBER	The R-factor integer * 1000 this is used in the MOS score compilation on the destination call leg.
RX1000FactorOnSrcLeg	type="uint32"	type="int4"	NUMBER	The R-factor integer * 1000 this is used in the MOS score compilation on the destination call leg.
mos-fmt-on-src-leg	type="String"	type="name"	VARCHAR2 (256)	The formatted MOS calculation on the inbound call leg.
mos-fmt-dest-leg	type="String"	type="name"	VARCHAR2 (256)	The formatted MOS calculation on the outbound call leg.
call-type	type="String"	type="name"	VARCHAR2 (256)	The type of call, such as IV for Inbound Voice. This is the only current value for this field.
disconnect-error-type	type="String"	type="name"	VARCHAR2 (256)	The type of error that caused the disconnection.
ani	type="String"	type="name"	VARCHAR2 (256)	The caller ID for the ANI after any manipulation by the OS-E.
call-source-regid	type="String"	type="name"	VARCHAR2 (256)	The server name if available, or user portion of the From: URI.
call-dest-regid	type="String"	type="name"	VARCHAR2 (256)	The server name if available, or user portion of the To: URI.
new-ani	type="String"	type="name"	VARCHAR2 (256)	The caller ID for the ANI after any manipulation by the OS-E.
cdr-type	type="String"	type="name"	VARCHAR2 (256)	The call record type, either START or STOP.
hunting-attempts	type="uint32"	type="int4"	NUMBER	The number of times the OS-E used the arbiter to select a dial-plan and a failure occurred (including subsequent attempts).
call-pdd	type="uint32"	type="int4"	NUMBER	The post dial delay between the initial INVITE and the 180/183 RINGING.; calculated in msec.
call-source-realm-name	type="String"	type="name"	VARCHAR2 (256)	The source domain name from which the call was received.

CDR field	MS-SQL data types	PostgreSQL data types	Oracle data types	Description
call-dest-realm-name	type="String"	type="name"	VARCHAR2 (256)	The destination domain name to which the call was forwarded.
call-dest-cr-name	type="String"	type="name"	VARCHAR2 (256)	The name of the dial plan that forwarded the call.
in_peer_src	type="IPPort"	type="name"	VARCHAR2 (256)	The IP address and port of the source phone that contacted the OS-E over an inbound call leg.
in_peer_dst	type="IPPort"	type="name"	VARCHAR2 (256)	The IP address and port of the destination phone to which the OS-E forwarded the inbound call leg.
out_peer_src	type="IPPort"	type="name"	VARCHAR2 (256)	The IP address and port of the responding destination phone from which an outbound call leg was returned to the OS-E.
out_peer_dst	type="IPPort"	type="name"	VARCHAR2 (256)	The IP address and port of the destination phone to which the OS-E forwarded the outbound (return) call leg.
in_anchor_dst	type="IPPort"	type="name"	VARCHAR2 (256)	The IP address and port at the OS-E where the inbound call leg was received from the source peer.
in_anchor_src	type="IPPort"	type="name"	VARCHAR2 (256)	The IP address and port at the OS-E where the inbound call leg was forwarded to the destination peer.
out_anchor_dst	type="IPPort"	type="name"	VARCHAR2 (256)	The IP address and port at the OS-E where the outbound (responding) call leg was received from the destination peer.
out_anchor_src	type="IPPort"	type="name"	VARCHAR2 (256)	The IP address and port at the OS-E where the outbound call leg was forwarded back to the source peer.
called-party-after-src-calling-plan	type="String"	type="name"	VARCHAR2 (256)	The called party number after any manipulation on leg 1, but before any manipulation on leg 2.

CDR field	MS-SQL data types	PostgreSQL data types	Oracle data types	Description
last-status-message	type="uint16"	type="int4"	NUMBER	An integer indicating SIP message type last status message (omitting "200 OK") and therefore call progress.
last-pkt-timestamp-on-dest-leg	type="Time"	type="timestamp"	TIMESTAMP	The time of the last media packet on the destination leg.
last-pkt-timestamp-on-src-leg	type="Time"	type="timestamp"	TIMESTAMP	The time of the last media packet on the source leg.
setup-time-integer	type="uint64"	type="int8"	NUMBER	The call setup time indicated as an integer.
incoming-uri-stripped	type="String"	type="name"	VARCHAR2 (256)	The stripped down version of the incoming request URI.
dnis	type="String"	type="name"	VARCHAR2 (256)	Dialed Number Identification Service
newDnis	type="String"	type="name"	VARCHAR2 (256)	New Dialed Number Identification Service
max-jitter-on-src-leg	type="uint32"	type="int4"	NUMBER	The maximum jitter on the source leg.
max-jitter-on-dst-leg	type="uint32"	type="int4"	NUMBER	The maximum jitter on the destination leg.
max-latency-on-src-leg	type="uint32"	type="int4"	NUMBER	The maximum latency on the source leg.
max-latency-on-dst-leg	type="uint32"	type="int4"	NUMBER	The maximum latency on the destination leg.
customData	type="String"	type="name"	VARCHAR2 (256)	Contains user-specified information configured in the accounting-data session-config object. For more information see the CDR Custom Data Fields and Reserved Keywords section in the Oracle Communications OS-E Objects and Properties Reference Guide.
CreationTimestamp	type="Time" key="index"	type="timestamp"	TIMESTAMP	The time that the CDR was initially written to the target.

Sending CDRs to External Databases

When sending accounting CDRs to external databases, values that are unsigned 32-bit integers are stored as signed 32 bit integers in the database record. If the value of the field is larger than 2147483647 and retrieved as an integer, the value is stored as a negative number.

To decode the negative number, add 2^{32} or 4294967296 to the value.

The following columns are affected:

- Duration
- PacketsReceivedOnSrcLeg
- PacketsLostOnSrcLeg
- PacketsDiscardedOnSrcLeg
- PdvOnSrcLeg
- MaxJitterOnSrcLeg
- LatencyOnSrcLeg
- MaxLatencyOnSrcLeg
- PacketsReceivedOnDestLeg
- PacketsLostOnDestLeg
- PacketsDiscardedOnDestLeg
- PdvOnDestLeg
- MaxJitterOnDestLeg
- LatencyOnDestLeg
- MaxLatencyOnDestLeg
- Rx1000FactorOnDestLeg
- Rx1000FactorOnSrcLeg
- huntingAttempts
- callPDD

Appendix C. DTMF

About This Appendix

This appendix provides information on the Dual-Tone Multi-Frequency (DTMF) telephone standard for digits on the telephone keypad once a call is established.

DTMF Overview

In a VoIP environment, there are three general methods to represent DTMF:

- **Signaling**—Sent in SIP or H323 messages
- **RFC-2833**—Sent over the media path as RTP with special four-byte payload defined in RFC-2833
- **Audio**—DTMF audio encoded in the current CODEC for the stream (e.g. pcmu, ilbc, g729); sometimes referred to as *inband*

The signaling method guarantees delivery, but it typically introduces more delay as the messages are sent over the signaling path. The audio method may cause problems for endpoints connected by lower quality CODECs, such as g729.

DTMF has meaning for decimal digits between zero and 15. The following table shows the digits as sent in RFC-2833 packets, plus a decimal encoded value and the frequencies for the audio versions.

Digit	Decimal	Frequencies (Hz)
0	0	941 + 1336
1	1	697 + 1209
2	2	697 + 1336

Digit	Decimal	Frequencies (Hz)
3	3	697 + 1477
4	4	770 + 1209
5	5	770 + 1336
6	6	770 + 1477
7	7	852 + 1209
8	8	852 + 1336
9	9	852 + 1477
*	10	941 + 1209
#	11	941 + 1477
A	12	697 + 1633
B	13	770 + 1633
C	14	852 + 1633
D	15	941 + 1633

Signaling Overview

For SIP, DTMF information is carried in SIP INFO messages with either DTMF or DTMF-relay bodies. The DTMF has a single event without a duration. The DTMF-relay body contains a single event (called signal name) and a duration in milliseconds. Below is an example of a SIP INFO message with a DTMF-relay body:

```
INFO sip:1234@barry.companyXYZ.com:5060;maddr=172.26.0.235 SIP/2.0
Via: SIP/2.0/UDP
    172.30.1.6;rport;branch=z9hG4bKac1e0106000001848b4264300002e04000
    0010
Content-Length: 25
Call-ID: 4607306E-1D28-4CB7-8B95-79686373A772@172.30.1.6
Content-Type: application/dtmf-relay
CSeq: 2 INFO
From: "3933" <sip:3933@barry.companyXYZ.com>;tag=41800642118437
Max-Forwards: 70

To:
    <sip:1234@barry.companyXYZ.com>;tag=eb001aac-13c4-48b40b5d-1768641
    3-4220c819
User-Agent: SJphone/1.60.289a (SJ Labs)
Signal=5
Duration=1000
```

For H323, DTMF information is carried in either H245 or Q931 messages.

For H.245 MultimediaSystemControlMessages of type Indication, if the message decodes as UserInput, the OS-E supports both T_H245UserInputIndication_alphanumeric and T_H245UserInputIndication_signal/signalUpdate indications.

UserInputIndication_alphanumeric has the ability to carry a ASN1GeneralString representing the keys pressed. The OS-E currently supports UserInputIndicationSignal containing a string representing keys pressed plus a field indicating the press duration. SignalUpdate with a duration field updating the previous UIIndicationSignal is also supported. Below are some example decodes:

```
indication = {
  userInput = {
    alphanumeric = {
      "5"
    }
  }
}

indication = {
  userInput = {
    signal = {
      signalType = {
        "5"
      }
      duration = {
        4000
      }
    }
  }
}

indication = {
  userInput = {
    signalUpdate = {
      duration = {
        124
      }
    }
  }
}
```

Currently, the OS-E supports using signal and signalUpdate indications only for control or indication of the duration of DTMF. As covered in the H.245 specification, duration indicates the total duration of the tone (if known), or an initial estimate of the tone duration if the tone continues to be in progress at the time the signal is transmitted.

If duration is omitted, the receiver shall use an appropriate default based on the local configuration and network requirements. Duration shall be ignored in the case of a hookflash ("!") indication. signalUpdate revises the estimate of the total duration or declares the actual measured duration of the tone detected or to be generated. It should be transmitted so as to arrive well before the estimate that was previously sent in signal or signalUpdate expires. Otherwise, the revised duration will be ignored as the tone will have already been terminated by the receiver.

Note that it is not necessary to send signalUpdate if the total duration was indicated in signal.

Q.931 STATUS messages containing a Keypad facility Information Element are also supported by the OS-E. The Information Element contains one or more characters “entered by means of a terminal keypad” and is formatted as follows:

```
Q931: IE Keypad Facility (KF 0x2C)
      - Octet 1: IE type
          bits 8-1 2CH
      - Octets 3-n: mandatory
          bits 7-1 Keypad information (IA5 characters)
```

The OS-E tracing will show the Q.931 STATUS message decoded as follows:

```
{
  protocolDiscriminator = 8
  callReference = 4920
  from = originator
  messageType = 7b
  Keypad IE = {
    5
  }
}
```


a=sendrecv

For H323, the RFC-2833 packets are negotiated as describe in the Signaling Overview section.

Audio Overview

With DTMF sent as audio, it cannot be easily distinguished from normal audio packets since the only difference is within the RTP payload. The DTMF packets follow the same pattern as the normal RTP audio. The CODEC for the DTMF audio is negotiated as part of the signaling protocol used to establish the call.

DTMF Features

The OS-E performs conversions based on the session configuration **in-dtmf-preferences** (applies to the in-leg of a session) and **out-dtmf-preferences** (applies to the out-leg of a session) objects. DTMF methods are configured based on their assigned preference. Preference values are ordered in the range of 1-100 with a 1 being the highest preference, 100 being the lowest preference.

The OS-E sends the DTMF to an endpoint based on the most preferred method that an endpoint is capable of supporting. If more than one DTMF method is specified with the same preference, and the endpoint negotiates the ability to support each of the DTMF methods, then the DTMF method sent is the first DTMF found in the list of supported methods.

The OS-E supports the following DTMF types:

- Audio
- RFC-2833
- SIP INFO-DTMF-Relay
- SIP-INFO-DTMF
- SIP NOTIFY
- H.245 alphanumeric
- H.245 Signal

- Q.931

The OS-E can convert from signaling to RFC-2833 or audio, or RFC-2833 to signaling or audio. However, the OS-E does not monitor the audio stream for DTMF audio, so converting audio to signaling or RFC-2833 is not possible.

		Convert To		
		Signaling	RFC-2833	Audio
Convert From	Signaling	Note 1	yes	Note 2
	RFC-2833	yes	--	Notes 2 and 3.
	Audio	no	no	Note 3

- **Note 1**—Only done when converting between H323 and SIP
- **Note 2**—Ability to convert to audio was added in Release 3.5.0
- **Note 3**—Conversion of RFC-2833 or audio packets to audio can be done using transcoding (`session-config/media/decode-telephone-events=true`)

It is not advisable to change the `dtmf-translation drop-info` or `dtmf-rfc-2833` properties settings to *false*. This would mean that the DTMF event was applied to the media and signaling paths.

When converting to RFC-2833 or audio packets, the OS-E stops the current RTP stream and injects RTP packets on a random synchronization source (SSRC). During the time when injecting RTP packets, the OS-E drops the received packets going in the same direction, as they show up as DropForce in the kernel-rule-stats display. The random SSRC is used to denote a different source, where sequence numbers do not need to be a continuation of the sequence numbers on the other SSRC. Generally, the OS-E attempts to set the RTP marker bit in the first packet sent. The OS-E queues DTMF events if the previous event has not completed playing by the time the next event is received.

In addition, the OS-E has some settings that can be used to control the DTMF.

- When converting from RFC-2833, the OS-E listens for the end of an event so it can forward the duration of a call. The `dtmf-translation/timeout-rfc-2833` property is used to avoid long delays or missing the end. When the `dtmf-translation/timeout-rfc-2833` property is set to zero, the `vsp/dtmf-generation/digit-duration` setting is applied as the duration.

- When converting from RFC-2833, the OS-E protects against receiving DTMF event notification too frequently and is defaulted to 250 msecs, but can be controlled with **vsp/dtmf-generation/minimum-duration** property. Errors appear in the mstream class tracing on the SIP process when DTMF events are ignored.
- When converting from signaling, the **vsp/dtmf-generation/digit-duration** property is applied when no duration is provided.
- When converting to DTMF audio, the **vsp/dtmf-generation/digit-volume** property sets the audio level.

You also have the ability to control how the OS-E forwards DTMF tones in inbound and outbound calls via the **in-dtmf-settings** and **out-dtmf-settings**. These are configured under **session-config**.

To configure DTMF translation properties, access the **in-dtmf-settings** and **out-dtmf-settings** objects.

```
NNOS-E>config vsp
config vsp>config default-session-config
config default-session-config>config in-dtmf-settings
config in-dtmf-settings>set digit-volume -15
config in-dtmf-settings>set digit-duration 1000
config in-dtmf-settings>set min-digit-duration 50
config in-dtmf-settings>set max-digit-duration 5000
config in-dtmf-settings>set digit-duration-update 750
config in-dtmf-settings>set inter-digit-duration 300
config in-dtmf-settings>set pause-duration 4000
config in-dtmf-settings>set as-audio false
config in-dtmf-settings>return
config default-session-config>config out-dtmf-settings
config out-dtmf-settings>set digit-volume -15
config out-dtmf-settings>set digit-duration 1000
config out-dtmf-settings>set min-digit-duration 50
config out-dtmf-settings>set max-digit-duration 5000
config out-dtmf-settings>set digit-duration-update 750
config out-dtmf-settings>set inter-digit-duration 300
config out-dtmf-settings>set pause-duration 4000
config out-dtmf-settings>set as-audio false
config out-dtmf-settings>return
config default-session-config>return
```

SIP Configuration Example

In this example, endpoint A supports DTMF using SIP INFO dtmf-relay and endpoint B supports DTMF using RFC-2833. The following configuration ensures DTMF is translated properly if A initiates the call to B, or if B initiates the call to A:

```
Config session-config-pool
  Config entry From-A
    Config in-dtmf-preferences
      Set admin enabled
      Set preference sip-info-dtmf-relay 1
    Return
  Return
  Config entry To-A
    Config out-dtmf-preferences
      Set admin enabled
      Set preference sip-info-dtmf-relay 1
    Return
  Return
  Config entry From-B
    Config in-dtmf-preferences
      Set admin enabled
      Set preference rfc-2833 1
    Return
  Return
  Config entry To-B
    Config out-dtmf-preferences
      Set admin enabled
      Set preference rfc-2833 1
    Return
  Return
Return
```

H.323/SIP Configuration Example

In a configuration for a SIP/H.323 environment, the H.323 server configuration controls the behavior on the H.323 side, while “regular” session configuration controls the SIP side.

- For a SIP->H.323 call, only the **session-config/in-dtmf-translation** is applied.
- For an H.323-> SIP call, only the **session-config/out-dtmf-translation** is applied.

The **h323-server/sip-h323-dtmf-translate** is applied for each direction. In the example below, the intent is to go from SIP/RFC-2833 to a H.323/H.245 setup.

A ← SIP/RFC-2833 → OS-E ← H.323/H.245 → B

The following H.323 server configuration allows the OS-E to send DTMF as RFC-2833 packets on the SIP side and H.245 messages on the H.323 side:

```
sip-h323-dtmf-translate RFC2833 H245SIGNAL
```

Then, for SIP->H.323 calls the session-config should be:

```
config in-dtmf-translation
  set rfc-2833 info
```

And for H.323/SIP calls, the session-config should be:

```
config out-dtmf-translation
  set info rfc-2833
```

DTMF Troubleshooting

The steps below provide information on how to troubleshoot basic DTMF translation (translation not involving audio transcoding).

The **media-stream-dtmf** status provider shows DTMF events that are detected or injected by the OS-E. The injected DTMF events are events that are extracted from signaling messages and put into the media-stream as RFC-2833 packets. The detected events are RFC-2833 packets detected by the OS-E and changed into either signaling messages or media-stream audio.

```
SIP> show media-stream-dtmf
```

session-id	stream	call-leg	operation	event-id	volume	duration
0x4c318c62f2b8ba2	1	1	inject	5	20	2000
				6	20	2000
	2		detect	5	20	180
				9	20	100

Troubleshooting Conversion From RFC-2833

Perform the following steps:

1. Check that DTMF events are detected.

Session Detected Event

Once it is determined that the DTMF events were detected, you will need to determine whether or not DTMF events are sent.

With an active call, use the **show media-stream-dtmf** command to see if DTMF events were detected.

```
SIP> show media-stream-dtmf
```

session-id	stream	call-leg	operation	event-id	volume	duration
-----	-----	-----	-----	-----	-----	-----
0x4c318c62f2b8ba2	1	2	detect	5	20	180
				9	20	100

In the above example, two DTMF events were detected on the session. The first event is 5 with a duration of 180 msec, and the second event is with a duration of 100 msec. Since these DTMF events are detected, next determine whether or not the DTMF events are sent.

With the **trace mstream** action enabled on the SIP process, you would see an output that is similar to the text below when a DTMF event is detected:

```
mstream[info]: mstream_res_process_kernel_event:4727 event=DTMFEvent
  on sess-id=0x04c30638d1747aeb, stream=1, call-leg=2, rule_index=0

mstream[info]: mstream_res_kernel_event_dtmf:4652 received DTMF event
  end: event_id=8, duration=8000
```

Check Kernel-Rules to Ensure Targets Are Configured

If no DTMF events are detected for a given session, examine the kernel-rules to determine if DTMF detection is properly configured. If a Modify target is not present in the kernel rule, then the OS-E does not attempt to detect DTMF.

The following show convert RFC-2833 in one direction:

```
SIP> show kernel-rule rtp=true
```

source	dest	Prot	intf	info
-----	----	----	----	----

```
172.30.0.182:51574 172.26.0.235:21358 udp eth0 (promoted)
Rx:2169 Tx:2169 Drop:0 SourceTrack(
reset+promote+passInitial Probation Count:3)

SetDestSourceTuple(Dest 72.30.1.6:49154;Src 172.26.0.235:21538)
End

172.30.1.6:49154 172.26.0.235:21538 udp eth0
(promoted)
Rx:2166 Tx:2138 Drop:28
SourceTrack
(reset+promote+passInitial Probation Count:3)

Modify(DTMF pt=101,drop)

SetDestSourceTuple(Dest 172.30.0.182:51574;Src 172.26.0.235:21358)
End

SIP>
```

In the above example, DTMF is detected by examining the **RTP payload-type=101**, as negotiated in SDP.

If the Modify target is setup and no DTMF events are detected, examining a packet capture may provide information. If you do not see a Modify target in the RTP kernel rules, consider the following conditions.

- Endpoint is not negotiating for RFC-2833 packets – Examine the call logs to determine if the endpoint is negotiating for RFC-2833.
- Not hitting the intended dtmf-translation policy – The trace mstream debug action can be used to show the dtmf-translation that the OS-E is performing after the session has gone through all the policy manipulations.

Check Event-Log for Dropped DTMF Events

If some of the DTMF events for a given session are missing, open the event-log. The OS-E may drop DTMF events that arrive too close to each other to avoid flooding the signaling channels with DTMF messages. An event-log entry will be created for each ignored DTMF event.

The event-log entries will look as follows:

```
2009-01-08T18:11:32-05:00[error] 1:SIP[media] Session 4c3275b93e23020,
leg 0 dropped DTMF event-id=3: last event only started 181 msec ago
(250 msec minimum)
```

The SIP process has a similar trace statement in the mstream trace class if a DTMF event is ignored. The minimum value can be adjusted using the `vsp/dtmf-generation/minimum-duration` property setting.

Check DTMF Counters

The events generated by the kernel are counted. The counts can be viewed with the following commands:

```
SIP> show media-stream-kernel-events
```

event	count
-----	-----
ruleAdded	20
ruleDeleted	16
ruleChanged	15
srcIPChanged	20
verify	0
verifyPktSize	0
verifyPktRate	0
verifyRTP	0
verifyRTCP	0
mediaMonitor	0
SRTPKeySoftLimit	0
SRTPKeyHardLimit	0
SRTPKeyChange	0
SRTPEncryptKeyRollover	0
SRTPDecryptKeyRollover	0
SRTPKeyUnknown	0
DTMFEvent	2
RuleStatsScanned	0

```
SIP> show media-stream-session-kernel-events
```

session-id	event	count
-----	-----	-----
0x4c30638d1747aeb	ruleAdded	4
	ruleChanged	3
	srcIPChanged	4
	DTMFEvent	2

Check Packet Capture for RFC-2833 Packets

The media packet capture should show RFC-2833 packets. The RFC-2833 packets can be distinguished from the rest of the RTP by payload-type. The media packet capture can be examined with Ethereal/Wireshark or with the OS-E-recorded files.

Ethereal can be filtered to examine the RFC-2833 packets once the stream is identified as RTP. The relevant Ethereal filters are: 'rtpevent' or 'rtp.p_type==<payload-type>' where the payload-type is determined by the signaling. You should be able to see the incoming RFC-2833 packets, but no outgoing RFC-2833 packets (assuming the in-dtmf-translation or out-dtmf-translation/**drop-rfc-2833** property setting is set to true).

With the OS-E recording files, use the **rtp-stream** action to examine the packets within a received RTP stream.

```
Media> rtp-stream details /cxc_common/recorded/
sess-04c31d012c7740b6-0-1.xml

    1 Payload type=G729, SSRC=0xedd003f6, Seq=10652, Time=251645998,
      payload bytes=20

    2 Payload type=G729, SSRC=0xedd003f6, Seq=10653, Time=251646078,
      payload bytes=20

    3 Payload type=G729, SSRC=0xedd003f6, Seq=10654, Time=251646238,
      payload bytes=20

    4-30364 Payload type=G729, SSRC=0xedd003f6

30365 Payload type=G729, SSRC=0xedd003f6, Seq=41016, Time=256503998,
      payload bytes=20

**** Codec type change from G729(18) to telephone-event(101) ****

30366 Payload type=telephone-event, SSRC=0xedd003f6, Seq=41017,
      Time=256504238, Mark, payload bytes=4

30367 Payload type=telephone-event, SSRC=0xedd003f6, Seq=41018,
      Time=256504238, payload bytes=4

*** Codec type change from telephone-event(101) to G729(18) ****

30368 Payload type=G729, SSRC=0xedd003f6, Seq=41019, Time=256504398,
      payload bytes=20

30369-38457 Payload type=G729, SSRC=0xedd003f6
```

```
38458 Payload type=G729, SSRC=0xedd003f6, Seq=49109, Time=257798798,  
payload bytes=20
```

```
Media>
```

In the above recorded file, there are 2 DTMF packets in the RTP stream. The “analysis” argument (set to true by default) flags the codec change to RFC-2833 (telephone-event codec) and back to G729.

2. Check that DTMF events are sent.

Since RFC-2833 events may be converted to either signaling or audio, check the event log and any packet traces.

For Signaling Output Check Call-Logs For DTMF Messages

For RFC-2833 events sent as signaling, the call logs should contain the DTMF messages (for example, SIP INFO/dtmf-relay, H245). Additionally, the messages should appear in an Ethereal/Wireshark packet capture. If the DTMF messages do not appear in either of those places, examining traces may be useful in determining why the DTMF messages are not being injected into the signaling stream. For example, run the **trace * error** and **trace sip_traffic info** actions from the OS-E prompt.

For Audio Output Check SIP Tracing

When sending DTMF in the media-stream (audio or RFC-2833), it is best to use a combination of OS-E tracing and Ethereal. When injecting DTMF as audio, you should see the following traces inside the SIP process:

```
rtp[info]: rtp_ins_init_audio:749 created codec=pcmu (pt=0, ptime=20)  
rtp[info]: rtp_ins_init_dtmf:782 telephone-event: pt=101, ptime=20  
rtp[info]: rtp_ins_start:656 172.26.0.235:21494 ==>  
172.30.0.182:63646, curr-time=410840811, last-activity=410840811,  
SSRC=3555204444, Seq=62075, timestamp=2605331569
```

The OS-E sends the packets starting at the SSRC and Sequence number displayed in the traces. Regardless of the codec, the OS-E sets the RTP marker bit for the first packet in the injected audio.

For Audio Output, Check rtp-splice and rtp-stats

Some endpoints do not recognize the DTMF when it is not a continuation of the current RTP stream identified by SSRC, sequence numbers, and timestamps. In session-config/media, enable both the **rtp-splice** and **rtp-stats** properties.

The **rtp-stats** property tracks the current RTP information (SSRC, sequence number, and timestamp), so the OS-E can continue the same RTP stream when it injects DTMF (or other RTP audio). This may result in additional processing on each RTP packet after the normal audio stream is resumed, but is required for interoperability with some endpoints.

For Audio Output, Check Packet Capture For DTMF Audio In RTP Stream

It is useful to examine the Ethereal/Wireshark output. Without the session-config/media/**rtp-splice** property enabled, you should observe the RTP SSRC change during the DTMF event. When the OS-E injects audio, the RTP marker bit should be set ('rtp.marker==1') on the first packet regardless of **rtp-splice** setting. This provides an additional hint DTMF event start. If the codecs in use can be decoded by Ethereal, the call can be played using the following "Statistics" menu item -> "Voip Calls" menu item. Then, select the "Player" button and "Decode" the RTP.

Troubleshooting Conversion to RFC-2833 or Audio

Perform the following steps:

1. Check that DTMF events are detected.

The DTMF events may come in via RFC-2833 or signaling. Each method requires unique steps to examine whether the DTMF event is detected.

For RFC-2833 Input Check that the DTMF Events Are Detected

When converting from RFC-2833 packets to audio, follow the steps in the section "Troubleshooting Conversion From RFC-2833," to insure that the DTMF events are properly detected.

For Signaling Input Check Call-Logs for DTMF Messages

When using a signaling trigger, the call logs should show the signaling events. For active calls, you can use **trace sip_traffic info** action from the SIP process to see the SIP signaling message.

2. Check that DTMF events are sent.

When sending DTMF in the RTP stream (audio or RFC-2833), Oracle recommends that you use a combination of tracing and Ethereal.

Check SIP Tracing

When injecting DTMF into the RTP stream (as audio or RFC-2833), check the following traces inside the SIP process:

```
rtp[info]: rtp_ins_init_audio:749 created codec=pcmu (pt=0, ptime=20)
rtp[info]: rtp_ins_init_dtmf:782 telephone-event: pt=101, ptime=20

rtp[info]: rtp_ins_start:656 172.26.0.235:21494 ==>
172.30.0.182:63646, curr-time=410840811, last-activity=410840811,
SSRC=3555204444, Seq=62075, timestamp=2605331569
```

The OS-E sends the packets starting at the SSRC and Sequence number displayed in the traces. Regardless of the codec, Oracle sets the RTP marker bit for the first packet in the injected audio.

Check rtp-splice and rtp-stats

Some endpoints do not recognize the DTMF when it is not a continuation of the current RTP stream identified by SSRC, sequence numbers, and timestamps. In session-config/media, enable both the **rtp-splice** and **rtp-stats** properties.

The **rtp-stats** property tracks the current RTP information (SSRC, sequence number, and timestamp), so the OS-E can continue the same RTP stream when it injects DTMF (or other RTP audio). This may result in additional processing on each RTP packet after the normal audio stream is resumed, but is required for interoperability with some endpoints.

Check Packet Capture for DTMF in RTP Stream

It is useful to examine the Ethereal/Wireshark output. Without `session-config/media/rtp-splice` property enabled, you should observe the RTP SSRC change during the DTMF event. When the OS-E injects audio, the RTP marker bit should be set (`rtp.marker=1`) on the first packet regardless of `rtp-splice` setting. This provides an additional hint DTMF event start. If the codecs in use can be decoded by Ethereal, the call can be played using the following “Statistics” menu item -> “Voip Calls” menu item. Then, select the “Player” button and “Decode” the RTP.