Preface

This preface describes the objectives and organization of this guide and explains how to find additional information on related products and services. This preface contains the following sections:

- Guide Revision History, page 1
- Objectives, page 5
- Intended Audience, page 5
- Organization, page 6
- Related Documentation, page 8
- Conventions, page 9
- Configuration Guides, Command References, and Supplementary Resources, page 10
- Obtaining Documentation and Submitting a Service Request, page 16

Guide Revision History

The Guide Revision History records technical changes to this guide. The table shows the software release number and guide revision number for the change, the date of the change, and a brief summary of the change.

<table>
<thead>
<tr>
<th>Cisco IOS Release.</th>
<th>Part Number</th>
<th>Publication Date</th>
<th>Change Summary</th>
</tr>
</thead>
</table>
| Cisco IOS XE Release 3.11S | OL-19820-15 | November, 2013 | The following features were added:
|                   |             |                 | • Blended Transcoding |
| Cisco IOS XE Release 3.8S | OL-19820-14 | November, 2012 | The following features were added:
|                   |             |                 | • AMR-WB |
| Cisco IOS XE Release 3.7S | OL-19820-13 | July, 2012      | The following features were added:
|                   |             |                 | • H.248 Border Access Controller Support
|                   |             |                 | • IMS Rf Billing Interfaces |
| Cisco IOS XE Release 3.6S | OL-19820-12 | March 29, 2012  | The following features were added:
|                   |             |                 | • Common IP Address Media Bypass
<p>|                   |             |                 | • Via Header Passthrough |</p>
<table>
<thead>
<tr>
<th>Version</th>
<th>Document ID</th>
<th>Date</th>
<th>Features Added</th>
</tr>
</thead>
</table>
| Cisco IOS XE Release 3.5S | OL-19820-11 | November 28, 2011 | - Alarm-Related Enhancements  
- CAC-Related Enhancements  
- Call Log Correlation  
- Flexible Media Routing |
| Cisco IOS XE Release 3.4S | OL-19820-10 | July 25, 2011    | - Limiting Resource Usage  
- QoS Demarcation Enhancements  
- SDP Editing Using Script-Based Editors  
- SRTP Support for RTCP Multiplexed with RTP and for SSRC-Based Multiplexing |
| Cisco IOS XE Release 3.3S | OL-19820-09 | March 18, 2011   | - SIP Header Manipulation Enhancements  
- Support for H.239  
- Voice Transcoding Per Adjacency Statistics  
- Message, Policy, and Subscriber Statistics Enhancements  
- SPA DSP: Call Recovery  
- Flow Statistics QoS Enhancements  
- Selective Radius Billing  
- Alternative Contact Rewriting  
- BFCP Support  
- Limited H.323 ID Routing and Passthrough Support  
- Support to the Cisco ASR 1006 Series Router and Cisco ASR 1013 Series Router  
- Interchassis-Intrachassis Conversion |
The following features were added:

- SPA DSP Services
- Emergency and Security Enhancements
  - SIP trust model includes H.323 Interface
  - Emergency Call statistics
- SBC Calls Support using IPSec Tunnels
- ASR1001 Support
- XML based billing
- SIP Interworking Enhancements
  - Event Header in Publish Method
  - Source Number Editing during Number analysis
  - Privacy Service
  - Option Ping Enhancements
  - Multiple SBC media bypass
  - Add Expires Header to Register Message
  - Absence of Username Support in Request URI
- Analysis, Routing, and Policy Enhancements
  - Copy and Swap Procedure
  - Multiple CAC Averaging Periods
  - Administrative Domains
  - Blacklist Alerts
- Media Interworking Enhancements
  - MGX Assisted DTMF Interworking
  - Codec Preference and Re-Ordering
  - Per-Adjacency Codec String Interworking
  - Media Address Pool Support
- PKI High Availability Support
| Cisco IOS XE Release 3.1S | OL-19820-07 | July 30, 2010 | The following features were added:  
| | | | • IMS Rx and Diameter  
| | | | • ENUM Client feature  
| | | | • Customized System Error Messages  
| | | | • SRTP to RTP Interworking and SRTP Passthrough  
| | | | • Media Bandwidth Policy  
| | | | • SDP on 200 Invite  
| | | | • Memory Alerting  
| | | | • SIP Destination ID and SIP Source ID  
| | | | • Support for Asymmetric Payload Types  
| | | | • IP IPv6/VRF Feature  
| | | | • DTMF Method Interworking and ACCEPT Header Handling  
| | | | • CALEA IRI Interface Support feature  
| | | | • Redundant Peer Addresses  
| | | | • Per Subscriber Delete  
| Cisco IOS XE Release 2.6.2 | OL-19820-06 | July 08, 2010 | Endpoint information in PacketCable billing records was added.  
| Cisco IOS XE Release 2.6.1 | OL-19820-05 | April, 2010 | Adjacency information in PacketCable Billing Records was added.  
| Cisco IOS XE Release 2.6 | OL-19820-04 | February 26, 2010 | IPv6 support including IPv4 to IPv6 and IPv6 to IPv6 Interworking, Dynamic Codec Configuration, multiple audio and video codec support, H.323 support for Clear Channel calls, SIP-I Support and SIP Non-SDP Body Filtering, Unsignaled (granular-level) Secure Media, Configurable Mutual TLS Authentication per Interface, TLS Transport Parameter in Record-Router Header, Source Number Analysis, and Interoperability for SIP Authentication features were added.  
| Cisco IOS XE Release 2.5.1 | OL-19820-03 | January 27, 2010 | H.323 Extra TCS Codecs support was added.  

Objectives

This guide describes the Integrated Session Border Controller functions, features, restrictions, and configuration tasks for the Cisco ASR 1000 Series Aggregation Services Routers. It is not intended as a comprehensive guide to all of the software features that can be run using the Cisco ASR 1000 Series Routers, but only the Integrated Session Border Controller software specific to these Routers.

For information on general Cisco IOS software features that are also available on the Cisco ASR 1000 Series Routers, see the feature module or the technology guide for that software feature.

Intended Audience

This guide is intended for the following people:

- Experienced service provider administrators
- Cisco telecommunications management engineers
- Customers who use and manage Cisco ASR 1000 Series Routers
## Organization

This guide contains the following chapters and appendixes:

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Description</th>
</tr>
</thead>
</table>
| Part 1  | Basics | This part contains the following modules:  
- Using the Command-Line Interface in Cisco IOS XE Software  
- Cisco Unified Border Element (SP Edition) Overview  
- Configuring Cisco Unified Border Element (SP Edition)  
- Media Address Pools  
- Implementing Multi-VRF on Cisco Unified Border Element (SP Edition)  
- Implementing Adjacencies on Cisco Unified Border Element (SP Edition)  
- Implementing Cisco Unified Border Element (SP Edition) Policies  
- Call Duration Monitoring  
- IP Realm Support  
- Managing Emergency Calls |
| Part 2  | Service | This part contains the following modules:  
- Unexpected Source Address Alerting  
- DoS Prevention and Dynamic Blacklisting |
| Part 3  | Dual Tone Multifrequency (DTMF) | This part contains the following module:  
- Implementing Interworking DTMF |
| Part 4  | Redundancy-High Availability | This part contains the following modules:  
- Cisco Unified Border Element (SP Edition) Redundancy—High Availability Support  
- Interchassis High Availability |
| Part 5  | Media | This part contains the following modules:  
- Fax Support  
- Codec Handling, page 1  
- SDP Bandwidth Field Features  
- SDP Handling  
- Flexible Media Routing |
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Description</th>
</tr>
</thead>
</table>
| Part 6  | Session Initiation Protocol (SIP) | This part contains the following modules:  
- Inherit Profiles for Non-IMS Adjacencies  
- Cisco Unified Border Element (SP Edition) Registration Features  
- SIP Message Manipulation  
- Signaling Congestion Handling  
- SIP IP-FQDN URI Translation  
- SIP Tel URI Support  
- SIP Timer  
- SIP Configuration Flexibility  
- SIP Renegotiation  
- 100rel Interworking Support  
- Customized System Error Messages  
- BFCP Support |
| Part 7  | H.323 | This part contains the following modules:  
- H.323 Support  
- H.323 to SIP Interworking  
- Support for H.239 |
| Part 8  | Billing | This part contains the following modules:  
- Implementing Billing on Cisco Unified Border Element (SP Edition)  
- Billing Support |
| Part 9  | Secure Real-Time Transport Protocol (SRTP) | This part contains the following module:  
- Secure Media and SRTP Passthrough |
| Part 10 | Quality of Service (QoS) | This part contains the following module:  
- Implementing QoS (Marking) |
| Part 11 | Transcoding | This part contains the following modules:  
- Implementing Transcoding  
- Cisco Unified Border Element (SP Edition)—SPA DSP Services |
| Part 12 | Management and Operations | This part contains the following modules:  
- Tracking Policy Failure Statistics  
- Implementing SNMP  
- Logging Support |
Related Documentation

This section refers you to other documentation that might also be useful as you configure your Cisco ASR 1000 Series Routers. The documentation listed below is available on Cisco.com.

For information on Cisco Unified Border Element (SP Edition) commands, see the Cisco Unified Border Element (SP Edition) Command Reference: Unified Model at:

For information on the Cisco Unified Border Element (SP Edition) distributed model, see the:

- Cisco Unified Border Element (SP Edition) Configuration Guide: Distributed Model at:

- Cisco Unified Border Element (SP Edition) Command Reference: Distributed Model at:

For information on the Cisco Unified Border Element (SP Edition) examples, see the Cisco Unified Border Element (SP Edition) Configuration Profile Examples at:
For other related command documentation, see the:

- Cisco IOS command reference books for the new Cisco ASR 1000 Series Router commands and commands in existing Cisco IOS features for this release at the following link:
  

- Command Lookup Tool for information about Cisco IOS commands in general or a Cisco IOS master commands list at the following link:

  http://tools.cisco.com/Support/CLILookup

For Quick Start guides and installation documentation for the Cisco ASR 1000 Series Router, see the hardware documentation that was provided as a part of this release at:


For information on new software features, see the:

- Cisco ASR 1000 Series Aggregation Services Routers Software Configuration Guide


- Cisco IOS XE release notes


For further information, see the Cisco ASR 1000 Series Aggregation Services Routers Documentation Roadmap at:


Documentation for the Cisco IOS XE configuration guides and feature modules can be found at:


### Conventions

This document uses the following conventions:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Indication</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>bold</strong> font</td>
<td>Commands and keywords and user-entered text appear in <strong>bold</strong> font.</td>
</tr>
<tr>
<td><em>italic</em> font</td>
<td>Document titles, new or emphasized terms, and arguments for which you supply</td>
</tr>
<tr>
<td></td>
<td>values are in <em>italic</em> font.</td>
</tr>
<tr>
<td>[   ]</td>
<td>Elements in square brackets are optional.</td>
</tr>
<tr>
<td>{x</td>
<td>y</td>
</tr>
<tr>
<td></td>
<td>vertical bars.</td>
</tr>
<tr>
<td>[x</td>
<td>y</td>
</tr>
<tr>
<td></td>
<td>vertical bars.</td>
</tr>
<tr>
<td>string</td>
<td>A nonquoted set of characters. Do not use quotation marks around the string</td>
</tr>
<tr>
<td></td>
<td>or the string will include the quotation marks.</td>
</tr>
<tr>
<td><strong>courier</strong> font</td>
<td>Terminal sessions and information the system displays appear in <strong>courier</strong></td>
</tr>
<tr>
<td></td>
<td>font.</td>
</tr>
<tr>
<td>&lt; &gt;</td>
<td>Nonprinting characters such as passwords are in angle brackets.</td>
</tr>
<tr>
<td>[   ]</td>
<td>Default responses to system prompts are in square brackets.</td>
</tr>
<tr>
<td>!, #</td>
<td>An exclamation point (!) or a pound sign (#) at the beginning of a line of</td>
</tr>
<tr>
<td></td>
<td>code indicates a comment line.</td>
</tr>
</tbody>
</table>
**Note**  
Means reader take note.

**Tip**  
Means the following information will help you solve a problem.

**Caution**  
Means reader be careful. In this situation, you might perform an action that could result in equipment damage or loss of data.

**Timesaver**  
Means the described action saves time. You can save time by performing the action described in the paragraph.

**Warning**  
Means reader be warned. In this situation, you might perform an action that could result in bodily injury.

## Configuration Guides, Command References, and Supplementary Resources

Table 1 lists, in alphabetical order, Cisco IOS XE software configuration guides and command references, including brief descriptions of the contents of the documents. The command references contain commands for both Cisco IOS software and Cisco IOS XE software, for all releases. The command references support many different software releases and platforms. Your Cisco IOS XE software release or platform may not support all these technologies.

Table 2 lists documents and resources that supplement the Cisco IOS XE software configuration guides and command references. These supplementary resources include release notes and caveats; master command lists; new, modified, removed, and replaced command lists; system messages; and the debug command reference.

For additional information about configuring and operating specific networking devices, and to access Cisco IOS documentation, go to the Product/Technologies Support area of Cisco.com at the following location:

http://www.cisco.com/go/techdocs

### Table 1  Cisco IOS XE Configuration Guides and Command References

<table>
<thead>
<tr>
<th>Configuration Guide and Command Reference Titles</th>
<th>Features/Protocols/Technologies</th>
</tr>
</thead>
<tbody>
<tr>
<td>- <strong>Cisco ASR 1000 Series Aggregation Services Routers</strong>&lt;br&gt;SIP and SPA Software Configuration Guide</td>
<td>Configuration and troubleshooting of SPA interface processors (SIPs) and shared port adapters (SPAs) that are supported on the Cisco ASR 1000 Series Router.</td>
</tr>
<tr>
<td>- <strong>Cisco ASR 1000 Series Aggregation Services Routers</strong>&lt;br&gt;Software Configuration Guide</td>
<td>Overview of software functionality that is specific to the Cisco ASR 1000 Series Aggregation Services Routers.</td>
</tr>
<tr>
<td>Configuration Guide and Command Reference Titles</td>
<td>Features/Protocols/Technologies</td>
</tr>
<tr>
<td>-------------------------------------------------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>• Cisco IOS XE Access Node Control Protocol Configuration Guide</td>
<td>Communication protocol between digital subscriber line access multiplexers (DSLAMs) and a broadband remote access server (BRAS).</td>
</tr>
<tr>
<td>• Cisco IOS Access Node Control Protocol Command Reference</td>
<td></td>
</tr>
<tr>
<td>• Cisco IOS XE Asynchronous Transfer Mode Configuration Guide</td>
<td>LAN ATM, multiprotocol over ATM (MPoA), and WAN ATM.</td>
</tr>
<tr>
<td>• Cisco IOS Asynchronous Transfer Mode Command Reference</td>
<td></td>
</tr>
<tr>
<td>• Cisco IOS XE Broadband Access Aggregation and DSL Configuration Guide</td>
<td>PPP over Ethernet (PPPoE).</td>
</tr>
<tr>
<td>• Cisco IOS Broadband Access Aggregation and DSL Command Reference</td>
<td></td>
</tr>
<tr>
<td>• Cisco IOS XE Carrier Ethernet Configuration Guide</td>
<td>IEEE 802.3ad Link Bundling; Link Aggregation Control Protocol (LACP) support for Ethernet and Gigabit Ethernet links and EtherChannel bundles; LACP support for stateful switchover (SSO), in service software upgrade (ISSU), Cisco nonstop forwarding (NSF), and nonstop routing (NSR) on Gigabit EtherChannel bundles; and IEEE 802.3ad Link Aggregation MIB.</td>
</tr>
<tr>
<td>• Cisco IOS Carrier Ethernet Command Reference</td>
<td></td>
</tr>
<tr>
<td>• Cisco IOS XE Configuration Fundamentals Configuration Guide</td>
<td>Autoinstall, Setup, Cisco IOS command-line interface (CLI), Cisco IOS file system (IFS), Cisco IOS web browser user interface (UI), basic file transfer services, and file management.</td>
</tr>
<tr>
<td>• Cisco IOS Configuration Fundamentals Command Reference</td>
<td></td>
</tr>
<tr>
<td>• Cisco IOS XE DECnet Configuration Guide</td>
<td>DECnet protocol.</td>
</tr>
<tr>
<td>• Cisco IOS DECnet Command Reference</td>
<td></td>
</tr>
<tr>
<td>• Cisco IOS XE Dial Technologies Configuration Guide</td>
<td>Asynchronous communications, dial backup, dialer technology, Multilink PPP (MLP), PPP, and virtual private dialup network (VPDN).</td>
</tr>
<tr>
<td>• Cisco IOS Dial Technologies Command Reference</td>
<td></td>
</tr>
<tr>
<td>• Easy Virtual Network Configuration Guide</td>
<td>Easy Virtual Network (EVN) is an IP-based virtualization technology that provides end-to-end virtualization of the network. With EVN, you can use a single IP infrastructure to provide separate virtual networks whose traffic paths remain isolated from each other.</td>
</tr>
<tr>
<td>• Easy Virtual Network Command Reference</td>
<td></td>
</tr>
<tr>
<td>• Cisco IOS XE High Availability Configuration Guide</td>
<td>A variety of high availability (HA) features and technologies that are available for different network segments (from enterprise access to service provider core) to facilitate creation of end-to-end highly available networks. Cisco IOS HA features and technologies can be categorized in three key areas: system-level resiliency, network-level resiliency, and embedded management for resiliency.</td>
</tr>
<tr>
<td>• Cisco IOS High Availability Command Reference</td>
<td></td>
</tr>
<tr>
<td>• Cisco IOS XE Intelligent Services Gateway Configuration Guide</td>
<td>Subscriber identification, service and policy determination, session creation, session policy enforcement, session life-cycle management, accounting for access and service usage, and session state monitoring.</td>
</tr>
<tr>
<td>• Cisco IOS Intelligent Services Gateway Command Reference</td>
<td></td>
</tr>
<tr>
<td>Configuration Guide and Command Reference Titles</td>
<td>Features/Protocols/Technologies</td>
</tr>
<tr>
<td>---------------------------------------------------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>• Cisco IOS XE Interface and Hardware Component Configuration Guide</td>
<td>LAN interfaces, logical interfaces, serial interfaces, virtual interfaces, and interface configuration.</td>
</tr>
<tr>
<td>• Cisco IOS Interface and Hardware Component Command Reference</td>
<td></td>
</tr>
<tr>
<td>• Cisco IOS XE IP Addressing Services Configuration Guide</td>
<td>IP addressing, Address Resolution Protocol (ARP), Network Address Translation (NAT), Domain Name System (DNS), Dynamic Host Configuration Protocol (DHCP), and Next Hop Address Resolution Protocol (NHRP).</td>
</tr>
<tr>
<td>• Cisco IOS IP Addressing Services Command Reference</td>
<td></td>
</tr>
<tr>
<td>• Cisco IOS IP Application Services Command Reference</td>
<td></td>
</tr>
<tr>
<td>• Cisco IOS XE IP Multicast Configuration Guide</td>
<td>Protocol Independent Multicast (PIM) sparse mode (PIM-SM), bidirectional PIM (bidir-PIM), Source Specific Multicast (SSM), Multicast Source Discovery Protocol (MSDP), Internet Group Management Protocol (IGMP), and Multicast VPN (MVPN).</td>
</tr>
<tr>
<td>• Cisco IOS IP Multicast Command Reference</td>
<td></td>
</tr>
<tr>
<td>• Cisco IOS XE IP Routing: BFD Configuration Guide</td>
<td>Bidirectional forwarding detection (BFD).</td>
</tr>
<tr>
<td>• Cisco IOS IP Routing: BGP Command Reference</td>
<td></td>
</tr>
<tr>
<td>• Cisco IOS IP Routing: EIGRP Command Reference</td>
<td></td>
</tr>
<tr>
<td>• Cisco IOS IP Routing: ISIS Command Reference</td>
<td></td>
</tr>
<tr>
<td>• Cisco IOS XE IP Routing: ODR Configuration Guide</td>
<td>On-Demand Routing (ODR).</td>
</tr>
<tr>
<td>• Cisco IOS IP Routing: ODR Command Reference</td>
<td></td>
</tr>
<tr>
<td>• Cisco IOS XE IP Routing: OSPF Configuration Guide</td>
<td>Open Shortest Path First (OSPF).</td>
</tr>
<tr>
<td>• Cisco IOS IP Routing: OSPF Command Reference</td>
<td></td>
</tr>
<tr>
<td>• Cisco IOS XE IP Routing: Protocol-Independent Configuration Guide</td>
<td>IP routing protocol-independent features and commands. Generic policy-based routing (PBR) features and commands are included.</td>
</tr>
<tr>
<td>• Cisco IOS IP Routing: Protocol-Independent Command Reference</td>
<td></td>
</tr>
<tr>
<td>• Cisco IOS IP Routing: RIP Command Reference</td>
<td></td>
</tr>
<tr>
<td>• Cisco IOS XE IP SLAs Configuration Guide</td>
<td>Cisco IOS IP Service Level Agreements (IP SLAs).</td>
</tr>
<tr>
<td>• Cisco IOS IP SLAs Command Reference</td>
<td></td>
</tr>
<tr>
<td>• Cisco IOS XE IP Switching Configuration Guide</td>
<td>Cisco Express Forwarding.</td>
</tr>
<tr>
<td>• Cisco IOS IP Switching Command Reference</td>
<td></td>
</tr>
<tr>
<td>Configuration Guide and Command Reference Titles</td>
<td>Features/Protocols/Technologies</td>
</tr>
<tr>
<td>-------------------------------------------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>• Cisco IOS XE LAN Switching Configuration Guide&lt;br&gt;• Cisco IOS LAN Switching Command Reference</td>
<td>VLANs and multilayer switching (MLS).</td>
</tr>
<tr>
<td>• Cisco IOS XE Multiprotocol Label Switching Configuration Guide&lt;br&gt;• Cisco IOS Multiprotocol Label Switching Command Reference</td>
<td>MPLS Label Distribution Protocol (LDP), MPLS Layer 2 VPNs, MPLS Layer 3 VPNs, MPLS Traffic Engineering (TE), and MPLS Embedded Management (EM) and MIBs.</td>
</tr>
<tr>
<td>• Cisco IOS XE NetFlow Configuration Guide&lt;br&gt;• Cisco IOS NetFlow Command Reference</td>
<td>Network traffic data analysis, aggregation caches, and export features.</td>
</tr>
<tr>
<td>• Cisco IOS XE Network Management Configuration Guide&lt;br&gt;• Cisco IOS Network Management Command Reference</td>
<td>Basic system management, system monitoring and logging, Cisco IOS Scripting with Tool Control Language (Tcl), Cisco networking services (CNS), Embedded Event Manager (EEM), Embedded Syslog Manager (ESM), HTTP, Remote Monitoring (RMON), and SNMP.</td>
</tr>
<tr>
<td>• Cisco IOS XE Novell IPX Configuration Guide&lt;br&gt;• Cisco IOS Novell IPX Command Reference</td>
<td>Novell Internetwork Packet Exchange (IPX) protocol.</td>
</tr>
<tr>
<td>• Cisco IOS XE Optimized Edge Routing Configuration Guide&lt;br&gt;• Cisco IOS Optimized Edge Routing Command Reference</td>
<td>Optimized edge routing (OER) monitoring and automatic route optimization and load distribution for multiple connections between networks.</td>
</tr>
<tr>
<td>• Cisco IOS XE Performance Routing Configuration Guide&lt;br&gt;• Cisco IOS Performance Routing Command Reference</td>
<td>Performance Routing (PfR) provides additional intelligence to classic routing technologies to track the performance of, or verify the quality of, a path between two devices over a WAN infrastructure in order to determine the best egress or ingress path for application traffic.</td>
</tr>
<tr>
<td>• Cisco IOS XE Quality of Service Solutions Configuration Guide&lt;br&gt;• Cisco IOS Quality of Service Solutions Command Reference</td>
<td>Class-based weighted fair queuing (CBWFQ), low latency queuing (LLQ), Modular Quality of Service (QoS) Command-Line Interface (CLI) (MQC), Network-Based Application Recognition (NBAR), priority queuing, Multilink PPP (MLP) for QoS, header compression, Resource Reservation Protocol (RSVP), weighted fair queuing (WFQ), and weighted random early detection (WRED).</td>
</tr>
<tr>
<td>• Cisco IOS Security Command Reference</td>
<td>Access control lists (ACLs); authentication, authorization, and accounting (AAA); firewalls; IP security and encryption; neighbor router authentication; network access security; public key infrastructure (PKI); RADIUS; and TACACS+.</td>
</tr>
<tr>
<td>Configuration Guide and Command Reference Titles</td>
<td>Features/Protocols/Technologies</td>
</tr>
<tr>
<td>-------------------------------------------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td><strong>Cisco IOS XE Security Configuration Guide: Secure Connectivity</strong></td>
<td>Internet Key Exchange (IKE) for IPsec VPNs; security for VPNs with IPsec; VPN availability features (reverse route injection, IPsec preferred peer, and real-time resolution for the IPsec tunnel peer); IPsec data plane features; IPsec management plane features; Public Key Infrastructure (PKI); Dynamic Multipoint VPN (DMVPN); Easy VPN; and Cisco Group Encrypted Transport VPN (GET VPN).</td>
</tr>
<tr>
<td><strong>Cisco IOS XE Security Configuration Guide: Securing the Data Plane</strong></td>
<td>Access Control Lists (ACLs); Firewalls: Context-Based Access Control (CBAC) and Zone-Based Firewall; Cisco IOS Intrusion Prevention System (IPS); Flexible Packet Matching; Unicast Reverse Path Forwarding (uRPF); Threat Information Distribution Protocol (TIDP) and TMS.</td>
</tr>
<tr>
<td><strong>Cisco IOS XE Security Configuration Guide: Securing User Services</strong></td>
<td>AAA (includes Network Admission Control [NAC]); Security Server Protocols (RADIUS and TACACS+); Secure Shell (SSH); Secure Access for Networking Devices (includes Autosecure and Role-Based CLI access); Lawful Intercept.</td>
</tr>
<tr>
<td><strong>Cisco IOS XE Service Advertisement Framework Configuration Guide</strong></td>
<td>Cisco Service Advertisement Framework.</td>
</tr>
<tr>
<td><strong>Cisco IOS Service Advertisement Framework Command Reference</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Cisco IOS XE VPDN Configuration Guide</strong></td>
<td>Multihop by Dialed Number Identification Service (DNIS), timer and retry enhancements for L2TP and Layer 2 Forwarding (L2F), RADIUS Attribute 82 (tunnel assignment ID), shell-based authentication of VPDN users, and tunnel authentication via RADIUS on tunnel terminator.</td>
</tr>
<tr>
<td><strong>Cisco IOS VPDN Command Reference</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Cisco IOS XE Wide-Area Networking Configuration Guide</strong></td>
<td>Frame Relay; L2VPN Pseudowire Redundancy; and Media-Independent PPP and Multilink PPP.</td>
</tr>
<tr>
<td><strong>Cisco IOS Wide-Area Networking Command Reference</strong></td>
<td></td>
</tr>
</tbody>
</table>
The Cisco Unified Border Element (Enterprise) brings a scalable option for enterprise customers. Running as a process on the Cisco ASR 1000 and utilizing the high-speed RTP packet processing path, the Cisco Unified Border Element (Enterprise) is used as an IP-to-IP gateway by enterprises and commercial customers to interconnect SIP and H.323 voice and video networks. The Cisco UBE (Enterprise) provides a network-to-network demarcation interface for signaling interworking, media interworking, address and port translations, billing, security, quality of service (QoS), and bandwidth management.

The Cisco Unified Border Element (Enterprise) is a session border controller (SBC) that is VoIP-enabled and deployed at the edge of networks. For Cisco IOS XE Release 2.3 and earlier releases, Cisco Unified Border Element (SP Edition) is supported only in the distributed mode. Operating in the distributed mode, the SBC is a toolkit of functions that can be used to deploy and manage VoIP services, such as signaling interworking, network hiding, security, and quality of service.

The Cisco Unified Border Element (SP Edition) is a highly scalable, carrier-grade session border controller (SBC) that is designed for service providers and that is generally deployed at the border of the enterprise or SP networks to enable the easy deployment and management of VoIP services. Cisco Unified Border Element (SP Edition) is integrated into Cisco routing platforms and can use a large number of router functions to provide a very feature-rich and intelligent SBC application. Formerly known as Integrated Session Border Controller, Cisco Unified Border Element (SP Edition) provides a network-to-network demarcation interface for signaling interworking, media interworking, address and port translations, billing, security, quality of service, call admission control, and bandwidth management.

For Cisco IOS XE Release 2.4 and later releases, Cisco Unified Border Element (SP Edition) can operate in two modes or deployment models: unified and distributed. The configuration guide documents the features in the unified mode.
Table 2 lists documents and resources that supplement the Cisco IOS XE software configuration guides and command references.

**Table 2  Cisco IOS XE Software Supplementary Documents and Resources**

<table>
<thead>
<tr>
<th>Document Title or Resource</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Cisco IOS Master Command List, All Releases</em></td>
<td>Alphabetical list of all the commands documented in all Cisco IOS XE software releases.</td>
</tr>
<tr>
<td><em>Cisco IOS Debug Command Reference</em></td>
<td>Alphabetical list of <code>debug</code> commands including brief descriptions of use, command syntax, and usage guidelines.</td>
</tr>
<tr>
<td>Cisco IOS XE system messages</td>
<td>List of Cisco IOS XE system messages and descriptions. System messages may indicate problems with your system, may be informational only, or may help diagnose problems with communications lines, internal hardware, or the system software.</td>
</tr>
<tr>
<td>Release notes and caveats</td>
<td>Information about new and changed features, system requirements, and other useful information about specific software releases; information about defects in specific Cisco IOS XE software releases.</td>
</tr>
<tr>
<td>MIBs</td>
<td>Files used for network monitoring. To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
<tr>
<td>RFCs</td>
<td>Standards documents maintained by the Internet Engineering Task Force (IETF) that Cisco IOS XE documentation references where applicable. The full text of referenced RFCs may be obtained at the following URL: <a href="http://www.rfc-editor.org/">http://www.rfc-editor.org/</a></td>
</tr>
</tbody>
</table>

**Obtaining Documentation and Submitting a Service Request**

For information on obtaining documentation, submitting a service request, and gathering additional information, see the monthly *What’s New in Cisco Product Documentation*, which also lists all new and revised Cisco technical documentation, at:


Subscribe to the *What’s New in Cisco Product Documentation* as a Really Simple Syndication (RSS) feed and set content to be delivered directly to your desktop using a reader application. The RSS feeds are a free service and Cisco currently supports RSS version 2.0.
Using the Command-Line Interface in Cisco IOS XE Software

This chapter provides basic information about the command-line interface (CLI) in Cisco IOS XE software and how you can use some of the CLI features. This document contains the following sections:

- Initially Configuring a Device, page 1-17
- Using the CLI, page 1-18
- Saving Changes to a Configuration, page 1-27
- Additional Information, page 1-28

For more information about using the CLI, see “Part 1: Using the Cisco IOS Command-Line Interface (CLI)” of Cisco IOS XE Configuration Fundamentals Configuration Guide.

For information about the software documentation set, see the “About Cisco IOS XE Software Documentation” document.

Initially Configuring a Device

Initially configuring a device varies by platform. For information about performing an initial configuration, see the hardware installation documentation that is provided with the original packaging of the product or go to the Product Support area of Cisco.com at http://www.cisco.com/go/techdocs.

After you have performed the initial configuration and connected the device to your network, you can configure the device by using the console port or a remote access method, such as Telnet or Secure Shell (SSH), to access the CLI or by using the configuration method provided on the device, such as Security Device Manager.

Changing the Default Settings for a Console or AUX Port

There are only two settings that you can change on a console port or an AUX port:

- Change the port speed with the config-register 0x command. Changing the port speed is not recommended. The well-known default speed is 9600.
- Change the behavior of the port; for example, by adding a password or changing the timeout value.

Note

The AUX port on the Route Processor (RP) installed in a Cisco ASR 1000 series router does not serve any useful customer purpose and should be accessed only under the advisement of a customer support representative.
Using the CLI

This section describes the following topics:

- Understanding Command Modes, page 1-18
- Using the Interactive Help Feature, page 1-21
- Understanding Command Syntax, page 1-22
- Understanding Enable and Enable Secret Passwords, page 1-23
- Using the Command History Feature, page 1-24
- Abbreviating Commands, page 1-25
- Using Aliases for CLI Commands, page 1-25
- Using the no and default Forms of Commands, page 1-26
- Using the debug Command, page 1-26
- Filtering Output Using Output Modifiers, page 1-26
- Understanding CLI Error Messages, page 1-27

Understanding Command Modes

The CLI command mode structure is hierarchical, and each mode supports a set of specific commands. This section describes the most common of the many modes that exist.

Table 1-1 lists common command modes with associated CLI prompts, access and exit methods, and a brief description of how each mode is used.

<table>
<thead>
<tr>
<th>Command Mode</th>
<th>Access Method</th>
<th>Prompt</th>
<th>Exit Method</th>
<th>Mode Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>User EXEC</td>
<td>Log in.</td>
<td>Router&gt;</td>
<td>Issue the <strong>logout</strong> or <strong>exit</strong> command.</td>
<td>• Change terminal settings.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Perform basic tests.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Display device status.</td>
</tr>
<tr>
<td>Privileged EXEC</td>
<td>From user EXEC mode,</td>
<td>Router#</td>
<td>Issue the <strong>disable</strong> command or the <strong>exit</strong> command to return to user EXEC mode.</td>
<td>• Issue <strong>show</strong> and <strong>debug</strong> commands.</td>
</tr>
<tr>
<td></td>
<td>issue the <strong>enable</strong></td>
<td></td>
<td></td>
<td>• Copy images to the device.</td>
</tr>
<tr>
<td></td>
<td>command.</td>
<td></td>
<td></td>
<td>• Reload the device.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Manage device configuration files.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Manage device file systems.</td>
</tr>
<tr>
<td>Global configuration</td>
<td>From privileged EXEC</td>
<td>Router(config)#</td>
<td>Issue the <strong>exit</strong> command or the <strong>end</strong> command to return to privileged EXEC mode.</td>
<td>Configure the device.</td>
</tr>
<tr>
<td></td>
<td>mode, issue the <strong>configure terminal</strong> command.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 1-1  CLI Command Modes (continued)

<table>
<thead>
<tr>
<th>Command Mode</th>
<th>Access Method</th>
<th>Prompt</th>
<th>Exit Method</th>
<th>Mode Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface configuration</td>
<td>From global configuration mode, issue the <code>interface</code> command.</td>
<td>Router(config-if)#</td>
<td>Issue the <code>exit</code> command to return to global configuration mode or the <code>end</code> command to return to privileged EXEC mode.</td>
<td>Configure individual interfaces.</td>
</tr>
<tr>
<td>Line configuration</td>
<td>From global configuration mode, issue the <code>line vty</code> or <code>line console</code> command.</td>
<td>Router(config-line)#</td>
<td>Issue the <code>exit</code> command to return to global configuration mode or the <code>end</code> command to return to privileged EXEC mode.</td>
<td>Configure individual terminal lines.</td>
</tr>
</tbody>
</table>
### ROM monitor

From privileged EXEC mode, issue the **reload** command. Press the **Break** key during the first 60 seconds while the system is booting.

The # symbol represents the line number and increments at each prompt.

Issue the **continue** command.

- Run as the default operating mode when a valid image cannot be loaded.
- Access the fall-back procedure for loading an image when the device lacks a valid image and cannot be booted.
- Perform password recovery when a CTRL-Break sequence is issued within 60 seconds of a power-on or reload event.

### Diagnostic

The router boots or enters diagnostic mode in the following scenarios. When a Cisco IOS XE process or processes fail, in most scenarios the router will reload.

- A user-configured access policy was configured using the **transport-map** command, which directed the user into diagnostic mode.
- The router was accessed using an RP auxiliary port.
- A break signal (Ctrl-C, Ctrl-Shift-6, or the send break command) was entered, and the router was configured to enter diagnostic mode when the break signal was received.

If a Cisco IOS XE process failure is the reason for entering diagnostic mode, the failure must be resolved and the router must be rebooted to exit diagnostic mode.

If the router is in diagnostic mode because of a transport-map configuration, access the router through another port or use a method that is configured to connect to the Cisco IOS XE CLI.

If the RP auxiliary port was used to access the router, use another port for access. Accessing the router through the auxiliary port is not useful for customer purposes.

- Inspect various states on the router, including the Cisco IOS XE state.
- Replace or roll back the configuration.
- Provide methods of restarting the Cisco IOS XE software or other processes.
- Reboot hardware, such as the entire router, an RP, an ESP, a SIP, a SPA, or other hardware components.
- Transfer files into or off of the router using remote access methods such as FTP, TFTP, and SCP.

<table>
<thead>
<tr>
<th>Command Mode</th>
<th>Access Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROM monitor</td>
<td>From privileged EXEC mode, issue the <strong>reload</strong> command. Press the <strong>Break</strong> key during the first 60 seconds while the system is booting.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Prompt</th>
<th>Exit Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>common # &gt;</td>
<td>Issue the <strong>continue</strong> command.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Diagnostic Command</th>
<th>Access Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>The router boots or enters diagnostic mode in the following scenarios. When a Cisco IOS XE process or processes fail, in most scenarios the router will reload.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Mode Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Run as the default operating mode when a valid image cannot be loaded.</td>
</tr>
<tr>
<td>• Access the fall-back procedure for loading an image when the device</td>
</tr>
<tr>
<td>lacks a valid image and cannot be booted.</td>
</tr>
<tr>
<td>• Perform password recovery when a CTRL-Break sequence is issued within</td>
</tr>
<tr>
<td>60 seconds of a power-on or reload event.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Prompt</th>
<th>Exit Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router(diag)#</td>
<td>If a Cisco IOS XE process failure is the reason for entering diagnostic mode, the failure must be resolved and the router must be rebooted to exit diagnostic mode. If the router is in diagnostic mode because of a transport-map configuration, access the router through another port or use a method that is configured to connect to the Cisco IOS XE CLI. If the RP auxiliary port was used to access the router, use another port for access. Accessing the router through the auxiliary port is not useful for customer purposes.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Mode Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Inspect various states on the router, including the Cisco IOS XE state.</td>
</tr>
<tr>
<td>• Replace or roll back the configuration.</td>
</tr>
<tr>
<td>• Provide methods of restarting the Cisco IOS XE software or other</td>
</tr>
<tr>
<td>processes.</td>
</tr>
<tr>
<td>• Reboot hardware, such as the entire router, an RP, an ESP, a SIP, a SPA,</td>
</tr>
<tr>
<td>or other hardware components.</td>
</tr>
<tr>
<td>• Transfer files into or off of the router using remote access methods</td>
</tr>
<tr>
<td>such as FTP, TFTP, and SCP.</td>
</tr>
</tbody>
</table>
EXEC commands are not saved when the software reboots. Commands that you issue in a configuration mode can be saved to the startup configuration. If you save the running configuration to the startup configuration, these commands will execute when the software is rebooted. Global configuration mode is the highest level of configuration mode. From global configuration mode, you can enter a variety of other configuration modes, including protocol-specific modes.

ROM monitor mode is a separate mode that is used when the software cannot load properly. If a valid software image is not found when the software boots or if the configuration file is corrupted at startup, the software might enter ROM monitor mode. Use the question symbol (?) to view the commands that you can use while the device is in ROM monitor mode.

```
rommon 1 >
alias               set and display aliases command
boot                boot up an external process
confreg             configuration register utility
cont                continue executing a downloaded image
context             display the context of a loaded image
cookie              display contents of cookie PROM in hex
.
.
.
rommon 2 >
```

The following example shows how the command prompt changes to indicate a different command mode:

```
Router> enable
Router# configure terminal
Router(config)# interface ethernet 1/1
Router(config-if)# ethernet
Router(config-line)# exit
Router(config)# end
Router#
```

A keyboard alternative to the `end` command is Ctrl-Z.

### Using the Interactive Help Feature

The CLI includes an interactive Help feature. Table 1-2 describes how to use the Help feature.

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>help</td>
<td>Provides a brief description of the Help feature in any command mode.</td>
</tr>
<tr>
<td>?</td>
<td>Lists all commands available for a particular command mode.</td>
</tr>
<tr>
<td>partial command?</td>
<td>Provides a list of commands that begin with the character string (no space between the command and the question mark).</td>
</tr>
<tr>
<td>partial command&lt;Tab&gt;</td>
<td>Completes a partial command name (no space between the command and &lt;Tab&gt;).</td>
</tr>
<tr>
<td>command?</td>
<td>Lists the keywords, arguments, or both associated with the command (space between the command and the question mark).</td>
</tr>
<tr>
<td>command keyword?</td>
<td>Lists the arguments that are associated with the keyword (space between the keyword and the question mark).</td>
</tr>
</tbody>
</table>
The following examples show how to use the help commands:

**help**

Router> help

Help may be requested at any point in a command by entering a question mark '?'. If nothing matches, the help list will be empty and you must backup until entering a '?' shows the available options.

Two styles of help are provided:
1. Full help is available when you are ready to enter a command argument (e.g. 'show ?') and describes each possible argument.
2. Partial help is provided when an abbreviated argument is entered and you want to know what arguments match the input (e.g. 'show pr?'.)

?  

Router# ?

Exec commands:
- access-enable         Create a temporary access-List entry
- access-profile        Apply user-profile to interface
- access-template       Create a temporary access-List entry
- alps                  ALPS exec commands
- archive               manage archive files

**partial command?**

Router(config)# zo?
zone  zone-pair

**partial command<Tab>**

Router(config)# we<Tab> webvpn

**command ?**

Router(config-if)# pppoe ?
- enable        Enable pppoe
- max-sessions  Maximum PPPOE sessions

**command keyword?**

Router(config-if)# pppoe enable ?
- group  attach a BBA group

---

**Understanding Command Syntax**

Command syntax is the format in which a command should be entered in the CLI. Commands include the name of the command, keywords, and arguments. Keywords are alphanumeric strings that are used literally. Arguments are placeholders for values that a user must supply. Keywords and arguments may be required or optional.

Specific conventions convey information about syntax and command elements. Table 1-3 describes these conventions.
Using the CLI

The following examples show syntax conventions:

```
Router(config)# ethernet cfm domain ?
   WORD domain name

Router(config)# ethernet cfm domain dname ?
   level

Router(config)# ethernet cfm domain dname level ?
   <0-7> maintenance level number

Router(config)# ethernet cfm domain dname level 7 ?
   <cr>

Router(config)# snmp-server file-transfer access-group 10 ?
   protocol protocol options
   <cr>

Router(config)# logging host ?
   Hostname or A.B.C.D IP address of the syslog server
   ipv6 Configure IPv6 syslog server
```

### Understanding Enable and Enable Secret Passwords

Some privileged EXEC commands are used for actions that impact the system, and it is recommended that you set a password for these commands to prevent unauthorized use. Two types of passwords, enable (not encrypted) and enable secret (encrypted), can be set. The following commands set these passwords and are issued in global configuration mode:

---

**Table 1-3 CLI Syntax Conventions**

<table>
<thead>
<tr>
<th>Symbol/Text</th>
<th>Function</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; &gt; (angle brackets)</td>
<td>Indicate that the option is an argument.</td>
<td>Sometimes arguments are displayed without angle brackets.</td>
</tr>
<tr>
<td>A.B.C.D.</td>
<td>Indicates that you must enter a dotted decimal IP address.</td>
<td>Angle brackets (&lt;&gt; ) are not always used to indicate that an IP address is an argument.</td>
</tr>
<tr>
<td>WORD (all capital letters)</td>
<td>Indicates that you must enter one word.</td>
<td>Angle brackets (&lt;&gt; ) are not always used to indicate that a WORD is an argument.</td>
</tr>
<tr>
<td>LINE (all capital letters)</td>
<td>Indicates that you must enter more than one word.</td>
<td>Angle brackets (&lt;&gt; ) are not always used to indicate that a LINE is an argument.</td>
</tr>
<tr>
<td>&lt;cr&gt; (carriage return)</td>
<td>Indicates the end of the list of available keywords and arguments, and also indicates when keywords and arguments are optional. When &lt;cr&gt; is the only option, you have reached the end of the branch or the end of the command if the command has only one branch.</td>
<td>—</td>
</tr>
</tbody>
</table>

---
Using the CLI

- `enable password`
- `enable secret password`

Using an enable secret password is recommended because it is encrypted and more secure than the enable password. When you use an enable secret password, text is encrypted (unreadable) before it is written to the config.text file. When you use an enable password, the text is written as entered (readable) to the config.text file.

Each type of password is case sensitive, can contain from 1 to 25 uppercase and lowercase alphanumeric characters, and can start with a number. Spaces are also valid password characters; for example, “two words” is a valid password. Leading spaces are ignored, but trailing spaces are recognized.

Note
Both password commands have numeric keywords that are single integer values. If you choose a number for the first character of your password followed by a space, the system will read the number as if it were the numeric keyword and not as part of your password.

When both passwords are set, the enable secret password takes precedence over the enable password.

To remove a password, use the `no` form of the commands: `no enable password` or `no enable secret password`.

For more information about password recovery procedures for Cisco products, see the following:


Using the Command History Feature

The command history feature saves the commands that you enter during a session in a command history buffer. The default number of commands saved is 10, but the number is configurable within the range of 0 to 256. This command history feature is particularly useful for recalling long or complex commands.

To change the number of commands saved in the history buffer for a terminal session, issue the `terminal history size` command:

```
Router# terminal history size num
```

A command history buffer is also available in line configuration mode with the same default and configuration options. To set the command history buffer size for a terminal session in line configuration mode, issue the `history` command:

```
Router(config-line)# history [size num]
```

To recall commands from the history buffer, use the following methods:

- Press Ctrl-P or the Up Arrow key—Recalls commands beginning with the most recent command. Repeat the key sequence to recall successively older commands.
- Press Ctrl-N or the Down Arrow key—Recalls the most recent commands in the history buffer after they have been recalled using Ctrl-P or the Up Arrow key. Repeat the key sequence to recall successively more recent commands.

Note
The arrow keys function only on ANSI-compatible terminals such as the VT100.
Issue the `show history` command in user EXEC or privileged EXEC mode—Lists the most recent commands that you entered. The number of commands that are displayed is determined by the setting of the `terminal history size` and `history` commands.

The command history feature is enabled by default. To disable this feature for a terminal session, issue the `terminal no history` command in user EXEC or privileged EXEC mode or the `no history` command in line configuration mode.

### Abbreviating Commands

Typing a complete command name is not always required for the command to execute. The CLI recognizes an abbreviated command when the abbreviation contains enough characters to uniquely identify the command. For example, the `show version` command can be abbreviated as `sh ver`. It cannot be abbreviated as `s ver` because `s` could mean `show`, `set`, or `systat`. The `sh v` abbreviation also is not valid because the `show` command has `vrrp` as a keyword in addition to `version`.

### Using Aliases for CLI Commands

To save time and the repetition of entering the same command multiple times, you can use a command alias. An alias can be configured to do anything that can be done at the command line, but an alias cannot move between modes, type in passwords, or perform any interactive functions.

Table 1-4 shows the default command aliases.

<table>
<thead>
<tr>
<th>Command Alias</th>
<th>Original Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>h</td>
<td>help</td>
</tr>
<tr>
<td>lo</td>
<td>logout</td>
</tr>
<tr>
<td>p</td>
<td>ping</td>
</tr>
<tr>
<td>s</td>
<td>show</td>
</tr>
<tr>
<td>u or un</td>
<td>undebug</td>
</tr>
<tr>
<td>w</td>
<td>where</td>
</tr>
</tbody>
</table>

To create a command alias, issue the `alias` command in global configuration mode. The syntax of the command is `alias mode command-alias original-command`. Following are some examples:

- `Router(config)# alias exec prt partition`—privileged EXEC mode
- `Router(config)# alias configure sb source-bridge`—global configuration mode
- `Router(config)# alias interface rl rate-limit`—interface configuration mode

To view both default and user-created aliases, issue the `show alias` command.

Using the no and default Forms of Commands

Most configuration commands have a no form that is used to reset a command to its default value or to disable a feature or function. For example, the ip routing command is enabled by default. To disable this command, you would issue the no ip routing command. To re-enable IP routing, you would issue the ip routing command.

Configuration commands may also have a default form, which returns the command settings to their default values. For commands that are disabled by default, using the default form has the same effect as using the no form of the command. For commands that are enabled by default and have default settings, the default form enables the command and returns the settings to their default values. To see what default commands are available on your system, enter default ? in the appropriate command mode of the command-line interface.

The no form is documented in the command pages of Cisco IOS command references. The default form is generally documented in the command pages only when the default form performs a function different than that of the plain and no forms of the command.

Command pages often include a “Command Default” section as well. The “Command Default” section documents the state of the configuration if the command is not used (for configuration commands) or the outcome of using the command if none of the optional keywords or arguments is specified (for EXEC commands).

Using the debug Command

A debug command produces extensive output that helps you troubleshoot problems in your network. These commands are available for many features and functions within Cisco IOS XE software. Some debug commands are debug all, debug aaa accounting, and debug mpls packets. To use debug commands during a Telnet session with a device, you must first enter the terminal monitor command. To turn off debugging completely, you must enter the undebug all command.


Caution

Debugging is a high priority and high CPU utilization process that can render your device unusable. Use debug commands only to troubleshoot specific problems. The best times to run debugging are during periods of low network traffic and when few users are interacting with the network. Debugging during these periods decreases the likelihood that the debug command processing overhead will affect network performance or user access or response times.

Filtering Output Using Output Modifiers

Many commands produce lengthy output that may use several screens to display. You can use output modifiers to filter this output to show only the information that you want to see.

The following three output modifiers are available:

- begin regular-expression—Displays the first line in which a match of the regular expression is found and all lines that follow.
- include regular-expression—Displays all lines in which a match of the regular expression is found.
• **exclude** *regular-expression*—Displays all lines except those in which a match of the regular expression is found.

To use one of these output modifiers, type the command followed by the pipe symbol (|), the modifier, and the regular expression that you want to search for or filter. A regular expression is a case-sensitive alphanumeric pattern. It can be a single character or number, a phrase, or a more complex string.

The following example illustrates how to filter output of the `show interface` command to display only lines that include the expression “protocol.”

```
Router# show interface | include protocol
```

```
FastEthernet0/0 is up, line protocol is up
Serial4/0 is up, line protocol is up
Serial4/1 is up, line protocol is up
Serial4/2 is administratively down, line protocol is down
Serial4/3 is administratively down, line protocol is down
```

### Understanding CLI Error Messages

You may encounter some error messages while using the CLI. Table 1-5 shows the common CLI error messages.

<table>
<thead>
<tr>
<th>Error Message</th>
<th>Meaning</th>
<th>How to Get Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>% Ambiguous command: “show con”</td>
<td>You did not enter enough characters for the command to be recognized.</td>
<td>Reenter the command followed by a space and a question mark (?). The keywords that you are allowed to enter for the command appear.</td>
</tr>
<tr>
<td>% Incomplete command.</td>
<td>You did not enter all the keywords or values required by the command.</td>
<td>Reenter the command followed by a space and a question mark (?). The keywords that you are allowed to enter for the command appear.</td>
</tr>
<tr>
<td>% Invalid input detected at “^” marker.</td>
<td>You entered the command incorrectly. The caret (^) marks the point of the error.</td>
<td>Enter a question mark (?) to display all the commands that are available in this command mode. The keywords that you are allowed to enter for the command appear.</td>
</tr>
</tbody>
</table>

For more system error messages, see *Cisco IOS XE System Messages*.

### Saving Changes to a Configuration

To save changes that you made to the configuration of a device, you must issue the `copy running-config startup-config` command or the `copy system:running-config nvram:startup-config` command. When you issue these commands, the configuration changes that you made are saved to the startup configuration and saved when the software reloads or power to the device is turned off or interrupted.
The following example shows the syntax of the `copy running-config startup-config` command:

```
Router# copy running-config startup-config
Destination filename [startup-config]?
```

You press Enter to accept the startup-config filename (the default), or type a new filename and then press Enter to accept that name. The following output is displayed indicating that the configuration was saved:

```
Building configuration...
[OK]
Router#
```

On most platforms, the configuration is saved to NVRAM. On platforms with a Class A flash file system, the configuration is saved to the location specified by the CONFIG_FILE environment variable. The CONFIG_FILE variable defaults to NVRAM.

## Additional Information

- “Part 1: Using the Cisco IOS Command-Line Interface (CLI)” of the *Cisco IOS XE Configuration Fundamentals Configuration Guide*
  

  or

  “Using Cisco IOS XE Software” chapter of the *Cisco ASR 1000 Series Aggregation Services Routers Software Configuration Guide*


- Cisco Product Support Resources
  

- Support area on Cisco.com (also search for documentation by task or product)
  

- Software Download Center (downloads; tools; licensing, registration, advisory, and general information) (requires Cisco.com user ID and password)
  

- Error Message Decoder, a tool to help you research and resolve error messages for Cisco IOS XE software
  
  [http://www.cisco.com/cgi-bin/Support/Errordecoder/index.cgi](http://www.cisco.com/cgi-bin/Support/Errordecoder/index.cgi)

- Command Lookup Tool, a tool to help you find detailed descriptions of Cisco IOS XE commands (choose *Select an index: IOS > Select a release: All IOS Commands*) (requires Cisco.com user ID and password)
  

- Output Interpreter, a troubleshooting tool that analyzes command output of supported `show` commands
  
  [https://www.cisco.com/pcgi-bin/Support/OutputInterpreter/home.pl](https://www.cisco.com/pcgi-bin/Support/OutputInterpreter/home.pl)
Cisco Unified Border Element (SP Edition) Overview

This chapter presents an overview of Cisco Unified Border Element (SP Edition) on the Cisco ASR 1000 Series Aggregation Services Routers—its signaling and media functions, unified and distributed deployment models, supported features, and supported MIBs.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).

Contents

- Cisco Unified Border Element (SP Edition) on the Cisco ASR 1000 Series Routers, page 2-1
- Supported MIBs, page 2-5

Cisco Unified Border Element (SP Edition) on the Cisco ASR 1000 Series Routers

Cisco Unified Border Element (SP Edition) enables direct IP-to-IP interconnect between multiple administrative domains for session-based services providing protocol interworking, security, and admission control and management. Cisco Unified Border Element (SP Edition) is a voice over IP (VoIP) device that sits on the border of a network and controls call admission to that network.

Cisco Unified Border Element (SP Edition) protects the interior of the network from excessive call load and malicious traffic. Cisco Unified Border Element (SP Edition) provides additional functions such as media bridging and billing services.

Cisco Unified Border Element (SP Edition) is integrated into the Cisco IOS Software and does not require any additional hardware to run.

The SBC service includes two functional areas:

- Signaling SBC function—Managed by the signaling border element (SBE), controls access of VoIP signaling messages to the core of the network, and manipulates the contents of these messages. It does this by acting as a Session Initiation Protocol (SIP) back-to-back user agent (B2BUA).
Media SBC function—Managed by the data border element (DBE), controls access of media packets to the network, provides differentiated services and quality of service (QoS) for different media streams, and prevents service theft. It does this by acting as a real-time transport protocol (RTP) proxy.

For Cisco IOS XE Release 2.4, Cisco Unified Border Element (SP Edition) can operate in two modes or deployment models:

- Unified—In the unified model, both the SBE and DBE logical entities co-exist on the same network element. In this model, the signaling entity controls the media local to the router. Simply put, the SBE handles the SIP and H.323 packets and the DBE handles the RTP and RTCP packets.

- Distributed—In the distributed model, the SBE and the DBE entities reside on two different network elements. Logically, each of the SBE entities controls multiple DBE elements, and each DBE could be controlled by multiple SBE entities. The SBE interacts with the DBE entities using a session controller interface (SCI). The SCI interface supports the H.248 protocol.

In this model, the bearer always flows through the DBE, and the SBE participates only in the signaling flow. This model is typically used in conjunction with a third-party SBE that supports the DBE H.248 profile.

---

**Note**

It is important to note that the DBE configuration is still required when running in the unified model because the DBE configuration provides the information necessary for the RTP media to flow.

**Note**

For Cisco IOS XE Release 2.3 and earlier, the SBC supports only DBEs in the distributed model.
Figure 2-1 illustrates the unified mode. Figure 2-2 illustrates the relationships between SBEs, DBEs, and other network elements.

**Figure 2-1** Relationships Between SBEs/DBEs and Other Network Elements in the Unified Model
In this diagram, adjacencies 1, 2, and 3 have been associated with the respective DBE locations. The first (double line) call comes in over adjacency 1 and is routed over adjacency 3. The second (single line) call comes over adjacency 2 and is routed over adjacency 3. The SBE picks a DBE from the appropriate location to process the call media.
Cisco Unified Border Element (SP Edition) on the Cisco ASR 1001 Series Routers

Table 2-1 list the scaling and performance that is supported on the Cisco ASR 1001 Series Routers.

<table>
<thead>
<tr>
<th>Platform</th>
<th>HT=180</th>
<th>RP CPU</th>
<th>QFP CPU</th>
<th>Degradation CPS %</th>
<th>Reasons for Congestion</th>
<th>Memory Setup</th>
<th>Throughput Value</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASR 1001 1 RU</td>
<td>CPS=60</td>
<td>33%</td>
<td>40%</td>
<td>NA</td>
<td>Memory</td>
<td>Default</td>
<td>5000K</td>
<td>NA</td>
</tr>
<tr>
<td>ASR 1001 1 RU</td>
<td>CPS=57</td>
<td>66%</td>
<td>78%</td>
<td>NA</td>
<td>Memory</td>
<td>Default</td>
<td>2500L</td>
<td>NA</td>
</tr>
</tbody>
</table>

Supported MIBs

The following MIBs are supported Cisco IOS XE Release 2.4 and later for the SBC on the Cisco ASR 1000 Series Router:

- CISCO-SESSION-BORDER-CONTROLLER-EVENT-MIB
- CISCO-SESSION-BORDER-CONTROLLER-CALL-STATS-MIB

For more information about MIB support on a Cisco ASR 1000 Series Routers, refer to the Cisco ASR 1000 Series Aggregation Services Routers MIB Specifications Guide at:


To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at:


To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you.
This chapter describes how to configure the data border element (DBE) and signaling border element (SBE) for Cisco Unified Border Element (SP Edition).

Note that the DBE configuration is still required when running in the unified model because the DBE configuration provides the information necessary for the RTP media to flow.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).

For a complete description of the commands used in this chapter, refer to the Cisco Unified Border Element (SP Edition) Command Reference: Unified Model at:


For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

**Configuring Unified Model**

This section contains the following information on configuring the unified model:

- Configuring SBE in the Unified Model, page 3-1
- Memory Alerting, page 3-10
- Configuring Memory Alerting, page 3-11
- Configuring DBE in the Unified Model, page 3-12
- Image Upgrade Procedure for Cisco Unified Border Element (SP Edition), page 3-14

**Configuring SBE in the Unified Model**

This section describes how to configure a SBE on a Cisco ASR 1000 Series Routers:

**Prerequisites**

- In the unified mode, you must configure the SBE before the DBE.
You need to configure blacklisting to override default blacklisting thresholds when the SBE is configured and before you start using Cisco Unified Border Element (SP Edition). See the Dynamic Blacklisting Behavior, page 12-5 for configuration information.

When running Cisco Unified Border Element (SP Edition) with 500 or more active calls, configure the huge buffer size to 65535 bytes with the `buffer huge size 65535` command. The increased buffer size is required because by default Cisco IOS software sets the “huge” buffer size to be 18084 bytes, which is not large enough for audit responses when there are more than 500 active calls.

**Configuration Tip**

We strongly recommend you use different addresses for signaling and media addresses to avoid scenarios where reservation for media port range can prevent call signaling packets from reaching the route processor (RP). In this scenario, if the SBC attempts to receive a call using a port that has been reserved by the SBC for media, packets will be dropped, rather than forwarded to the RP. This type of scenario is more likely to occur for H.323 and SIP calls using TCP transport.

**SUMMARY STEPS**

1. configure
2. sbc `sbc-name`
3. sbe
4. adjacency sip `adjacency-name`
5. signaling-address ipv4 `ipv4_IP_address`
6. signaling-port `port_num`
7. remote-address ipv4 `ip-address ip-mask`
8. signaling-peer `peer_name`
9. signaling-peer-port `port_num`
10. attach
11. exit
12. adjacency sip `adjacency-name`
13. signaling-address ipv4 `ipv4_IP_address`
14. signaling-port `port_num`
15. remote-address ipv4 `ip-address ip-mask`
16. signaling-peer `peer_name`
17. signaling-peer-port `port_num`
18. attach
19. call-policy-set `policy-set-id`
20. first-call-routing-table `table-name`
21. rtg-src-adjacency-table `table-id`
22. entry `entry-id`
23. action
24. dst-adjacency `target-adjacency`
25. match-adjacency key
26. exit
27. entry entry-id
28. action
29. dst-adjacency target-adjacency
30. match-adjacency key
31. complete
32. active-call-policy-set policy-set-id
33. activate
34. end
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc sbc-name</td>
<td>Creates the SBC service on Cisco Unified Border Element (SP Edition) and enters into SBC configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# sbc mySbc</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Enters the mode of the signaling border element (SBE) function of the SBC.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> adjacency sip adjacency-name</td>
<td>Enters the mode of an SBE SIP adjacency. Use the <code>adjacency-name</code> argument to define the name of the service.</td>
</tr>
<tr>
<td><strong>Note</strong> A functional SBC needs a minimum of two adjacencies configured.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> signaling-address ipv4 ipv4_IP_address</td>
<td>Specifies the local IPv4 signaling address of the SIP adjacency.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-adj-sip)# signaling-address ipv4 88.103.29.100</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> signaling-port port_num</td>
<td>Specifies the local signaling port of the SIP adjacency.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-adj-sip)# signaling-port 5060</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> remote-address ipv4 ip-address ip-mask</td>
<td>Restricts the set of remote signaling peers contacted over the adjacency to those with the given IP address prefix.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-adj-sip)# remote-address 200.200.200.0 255.255.255.0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> signaling-peer peer_address</td>
<td>Specifies the remote signaling peer for the SIP adjacency to use.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-adj-sip)# signaling-peer 200.200.200.118</td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong> signaling-peer-port port_num</td>
<td>Specifies the remote signaling-peer port for the SIP adjacency to use.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-adj-sip)# signaling-peer-port 5060</td>
<td></td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
</tr>
<tr>
<td>-------</td>
<td>------------------------------------</td>
</tr>
<tr>
<td>10</td>
<td>attach</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-adj-sip)#</td>
</tr>
<tr>
<td>11</td>
<td>exit</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-adj-sip)# exit</td>
</tr>
<tr>
<td>12</td>
<td>adjacency sip adjacency-name</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe)# adjacency sip Core</td>
</tr>
<tr>
<td>13</td>
<td>signaling-address ipv4 ipv4_IP_address</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-adj-sip)# signaling-address ipv4 88.103.33.100</td>
</tr>
<tr>
<td>14</td>
<td>signaling-port port_num</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-adj-sip)# signaling-port 5060</td>
</tr>
<tr>
<td>15</td>
<td>remote-address ipv4 ip-address ip-mask</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-adj-sip)# remote-address 200.200.200.0 255.255.255.0</td>
</tr>
<tr>
<td>16</td>
<td>signaling-peer peer_address</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-adj-sip)# signaling-peer 200.200.200.118</td>
</tr>
<tr>
<td>17</td>
<td>signaling-peer-port port_num</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-adj-sip)# signaling-peer-port 5060</td>
</tr>
<tr>
<td>18</td>
<td>attach</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-adj-sip)# attach</td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Step 19</th>
<th>Call-Policy-Set policy-set-id</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td><code>Router(config-sbc-sbe-callpolicy)# call-policy-set 1</code></td>
</tr>
<tr>
<td>Purpose</td>
<td>Enters the mode of routing policy set configuration within an SBE entity, creating a new policy set, if necessary.</td>
</tr>
<tr>
<td>Note</td>
<td>There can only be one call policy set at any given time.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 20</th>
<th>First-Call-Routing-Table table-name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td><code>Router(config-sbc-sbe-callpolicy)# first-call-routing-table start-table</code></td>
</tr>
<tr>
<td>Purpose</td>
<td>Configures the name of the first policy table to process when performing the routing stage of policy for new-call events.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 21</th>
<th>Rtg-Src-Adjacency-Table table-id</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td><code>Router(config-sbc-sbe-callpolicy)# rtg-src-adjacency-table start-table</code></td>
</tr>
<tr>
<td>Purpose</td>
<td>Enters the configuration mode of a routing table (creating one if necessary) within the context of an SBE policy set whose entries match the source adjacency.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 22</th>
<th>Entry entry-id</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td><code>Router(config-sbc-sbe-callpolicy-rtgtable)# entry 1</code></td>
</tr>
<tr>
<td>Purpose</td>
<td>Enters the mode for configuring an entry in a routing table, creating the entry if necessary.</td>
</tr>
</tbody>
</table>

| Step 23 | Action [next-table goto-table-name | complete | reject] |
|---------|----------------------------------|
| Example: | `Router(config-sbc-sbe-callpolicy-rtgtable-entry)# action complete` |
| Purpose | Configures the action to take if this routing entry is chosen. Possible actions are: |
|         | • Set the name of the next routing table to process if the event matches this entry. This is done using the `next-table` keyword and the `goto-table-name` argument. |
|         | • Complete the action using the `complete` keyword. |
|         | • Reject the indicated action using the `reject` keyword. |

<table>
<thead>
<tr>
<th>Step 24</th>
<th>Dst-Adjacency target-adjacency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td><code>Router(config-sbc-sbe-callpolicy-rtgtable-entry)# dst-adjacency Core</code></td>
</tr>
<tr>
<td>Purpose</td>
<td>Configures the destination adjacency of an entry in a routing table.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 25</th>
<th>Match-Adjacency target-adjacency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td><code>Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# match-adjacency Access</code></td>
</tr>
<tr>
<td>Purpose</td>
<td>Configures the match value of an entry in a number analysis or routing table whose entries match against the source adjacency.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 26</th>
<th>Exit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td><code>Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# exit</code></td>
</tr>
<tr>
<td>Purpose</td>
<td>Exits mode for configuring an entry in a routing table and enters configuration mode of a routing table to create an entry.</td>
</tr>
</tbody>
</table>
## Step 27
**Command or Action**: `entry entry-id`

**Purpose**: Enters the mode for configuring an entry in a routing table, creating the entry if necessary.

**Example**:  
```
Router(config-sbc-sbe-callpolicy-rtgtable)# entry 2
```

## Step 28
**Command or Action**: `action [next-table goto-table-name | complete | reject]`

**Purpose**: Configures the action to take if this routing entry is chosen. Possible actions are:

- Set the name of the next routing table to process if the event matches this entry. This is done using the `next-table` keyword and the `goto-table-name` argument.
- Complete the action using the `complete` keyword.
- Reject the indicated action using the `reject` keyword.

**Example**:  
```
Router(config-sbc-sbe-callpolicy-rtgtable-entry)# action complete
```

## Step 29
**Command or Action**: `dst-adjacency target-adjacency`

**Purpose**: Configures the destination adjacency of an entry in a routing table.

**Example**:  
```
Router(config-sbc-sbe-callpolicy-rtgtable-entry)# dst-adjacency Access
```

## Step 30
**Command or Action**: `match-adjacency target-adjacency`

**Purpose**: Configures the match value of an entry in a number analysis or routing table whose entries match against the source adjacency.

**Example**:  
```
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# match-adjacency Core
```

## Step 31
**Command or Action**: `complete`

**Purpose**: Completes the CAC policy set when you have committed the full set.

**Example**:  
```
Router(config-sbc-sbe-cacpolicy-cactable-entry)# complete
```

## Step 32
**Command or Action**: `active-call-policy-set policy-set-id`

**Purpose**: Sets the active routing policy set within an SBE entity.

**Example**:  
```
Router(config-sbc-sbe)# active-call-policy-set 1
```

## Step 33
**Command or Action**: `activate`

**Purpose**: Initiates the SBC service.

**Example**:  
```
Router(config-sbc-sbe)# activate
```

## Step 34
**Command or Action**: `end`

**Purpose**: Exits SBC-DBE configuration mode and returns to EXEC mode.

**Example**:  
```
Router(config-sbc-sbe)# end
```
Modifying Existing Call Policy Set

A policy set is a group of policies that can be active on Cisco Unified Border Element (SP Edition) at any one time. If a policy set is active, then Cisco Unified Border Element (SP Edition) uses the rules defined within it to apply policy to events. Routing and number analysis are configured in a call policy set.

Only one policy set of each type can be active at any given time. You can switch the active policy set at any time. You cannot modify the currently active policy set without deactivating it. However you can modify policy sets that are not active. A policy set can be deleted, provided that it is not the active policy set.

To modify an existing call policy set, you must first deactivate it with the no active call-policy-set command and then execute a no complete command.

The following task deactivates the active call-policy-set.

**SUMMARY STEPS**

1. configure
2. sbc service-name
3. sbe
4. no active-call-policy-set policy-set-id
5. no complete
6. exit

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc sbc-name</td>
<td>Creates the SBC service on Cisco Unified Border Element (SP Edition) and enters into SBC configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# sbc mysbc</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Enters the mode of an SBE entity within an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> no active-call-policy-set policy-set-id</td>
<td>deactivates the active routing policy set within an SBE entity.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe)# no</td>
<td></td>
</tr>
<tr>
<td>active-call-policy-set 1</td>
<td></td>
</tr>
</tbody>
</table>
### Replacing an Existing Call Policy Set

Only one policy set of each type can be active at any given time. You can replace or switch the active policy set at any time. To do that, first deactivate the existing call policy set. Then activate the new call policy set for it to take effect.

#### SUMMARY STEPS

1. `configure`
2. `sbc service-name`  
   - Use the `service-name` argument to define the name of the service.
3. `sbe`
4. `no active-call-policy-set policy-set-id`
5. `active-call-policy-set policy-set-id`
6. `complete`
7. `exit`

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>configure</code></td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>sbc service-name</code></td>
<td>Enters the mode of an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# sbc myservice</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>sbe</code></td>
<td>Enters the mode of an SBE entity within an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc)# sbe</td>
<td></td>
</tr>
</tbody>
</table>
Configuring Unified Model

Memory Alerting

The Memory Alerting feature enables you to configure the number of active calls on an SBC based on the amount of free memory available on the device.

For example, an ASR1000 may support 5000 maximum active calls and support other features as well. In a scenario where the upper limit to the active calls is not configured, and other non-SBC features are also in use, there is a possibility that the SBC might use the system memory to a point that even the basic functions of the ASR1000 gets affected due to memory fragmentation or lack of memory.

The Memory Alerting feature enables you to configure thresholds and drop rates for various memory availability levels. This prevents the SBC from consuming memory for new calls or call registrations.

The Memory Alerting feature consists of four levels, Minor, Major, Critical, and Halt. The levels are defined based on the amount of processor memory available at a given time. Processor memory is checked after every ten new calls to determine the memory available.

You can configure the percentage of memory available to trigger each level, and define the number of calls to be rejected (0 to 10) from a set of 10 calls.

Table 3-1 represents the default percentages and drop rates.

<table>
<thead>
<tr>
<th>Level</th>
<th>Default Percentage of Memory Remaining</th>
<th>Number of Calls Rejected</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minor</td>
<td>&lt;= 25%</td>
<td>0 of 10</td>
</tr>
<tr>
<td>Major</td>
<td>&lt;= 20%</td>
<td>4 of 10</td>
</tr>
</tbody>
</table>

### Table 3-1 SBC Memory Alerting Levels, Default Memory Percentages, and Calls Rejected

<table>
<thead>
<tr>
<th>Step 4</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>no active-call-policy-set policy-set-id</td>
<td>Deactivates the active routing policy set within an SBE entity.</td>
</tr>
</tbody>
</table>

**Example:**

Router(config-sbc-sbe)# no active-call-policy-set 1

<table>
<thead>
<tr>
<th>Step 5</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>active-call-policy-set policy-set-id</td>
<td>Activates the new active routing policy set that is replacing the prior active routing policy set.</td>
</tr>
</tbody>
</table>

**Example:**

Router(config-sbc-sbe)# active-call-policy-set 6

<table>
<thead>
<tr>
<th>Step 6</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>complete</td>
<td>Completes the active routing policy set.</td>
</tr>
</tbody>
</table>

**Example:**

Router(config-sbc-sbe-sbepolicy-cacpolicy-cacable-entry)# no complete

<table>
<thead>
<tr>
<th>Step 7</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>exit</td>
<td>Exits the current mode of the configuration.</td>
</tr>
</tbody>
</table>

**Example:**

Router(config-sbc-sbe-sbe-rtgpolicy-rtgtable-entry)# exit
You cannot configure or modify the level Halt. If only 10 or lesser percentage of memory is available on the device, SBC stops accepting new calls.

Whenever a memory level change occurs, a message similar to the following is displayed on the console:

*July 2010 10:25:56.489:%SBC_COMP-3-MEMORY_ALERT: SBC memory congestion level has changed from CRITICAL to MINOR.
Usage: 883638296 of 1774290032 bytes.

Use the `[no] reject-threshold [level] memory [percentage] [reject rate]` command to configure the memory threshold and reject rate for new calls.

### Configuring Memory Alerting

This task configures the reject threshold and reject rate for new calls.

#### SUMMARY STEPS

1. configure
2. `sbc sbc-name`
3. `sbe`
4. `reject-threshold`
5. `end`
6. `show sbc sbc-name sbe call-stats reject-threshold`

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td></td>
</tr>
<tr>
<td><code>configure terminal</code></td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td></td>
</tr>
<tr>
<td><code>sbc sbc-name</code></td>
<td>Enables entry into the mode of an SBC service. Use the <code>sbc-name</code> argument to define the name of the SBC.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config)# sbc mySbc</code></td>
<td></td>
</tr>
</tbody>
</table>
Chapter 3  Configuring Cisco Unified Border Element (SP Edition)

Configuring Unified Model

This section describes how to configure a DBE on a Cisco ASR 1000 Series Routers in the unified model. The DBE configuration is still required when running in the unified model because the DBE configuration provides the information necessary for the RTP media to flow.

Prerequisites

When running Cisco Unified Border Element (SP Edition) with 500 or more active calls, configure the huge buffer size to 65535 bytes with the buffer huge size 65535 command to ensure the buffer is large enough for audit responses.

SUMMARY STEPS

1. configure
2. sbc sbc-name
3. media-address ipv4 A.B.C.D
4. activate
5. end

---

### Command or Action | Purpose
--- | ---
Step 3 sbe | Enables entry into the mode of an SBE entity within an SBC service.

**Example:**
`Router(config-sbc)# sbe`

Step 4 reject-threshold | Configures the memory threshold and reject rate for new calls.

**Example:**
`Router(config-sbc-sbe)# reject-threshold major memory 20 5`

Step 5 end | Enable exit from the config-sbc-sbe mode.

**Example:**
`Router(config-sbc-sbe)# end`

Step 6 show sbc sbc-name sbe call-stats reject-threshold | Shows the reject threshold details for the selected SBC.

**Example:**
`Router# show sbc mySbc sbe call-stats reject-threshold`
DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td>Enters global configuration mode.</td>
</tr>
</tbody>
</table>
| Example:
Router# configure terminal | |
| Step 2 sbc sbc-name | Creates the SBC service on Cisco Unified Border Element (SP Edition) and enters into SBC-DBE configuration mode. |
| Example:
Router(config)# sbc mySbc | |
| Step 3 media-address ipv4 {A.B.C.D} | Adds the IPv4 address which can be used by the DBE as a local media address. This address is the SBC virtual interface address. |
| Example:
Router(config-sbc)# media-address ipv4 1.1.1.1 | |
| Step 4 activate | Initiates the SBC service, for DBE and SBE. |
| Example:
Router(config-sbc)# activate | |
| Step 5 end | Exits SBC-DBE configuration mode and returns to Exec mode. |
| Example:
Router(config-sbc)# end | |

Configuring Cisco Unified Border Element (SP Edition) Unified Model: Example

The following is an example of a Cisco Unified Border Element (SP Edition) unified model configuration:

Router# show run sbc
Generating configuration....
sbc test
sbe
  adjacency sip Access
  signaling-address ipv4 88.103.29.100
  signaling-port 5060
  remote-address ipv4 200.200.200.0 255.255.255.0
  signaling-peer 200.200.200.118
  signaling-peer-port 5060
  attach
adjacency sip Core
  signaling-address ipv4 88.103.33.100
  signaling-port 5060
  remote-address ipv4 200.200.200.0 255.255.255.0
  signaling-peer 200.200.200.118
  signaling-peer-port 5060
  attach
call-policy-set 1
Configuring Memory Alerting: Example

The following example shows how to configure memory threshold and reject rate for new calls:

Router# configure
Router(config)# sbc mySbc
Router(config-sbc)# sbe
Router(config-sbc-sbe)# reject-threshold minor memory 30 0
Router(config-sbc-sbe)# reject-threshold major memory 20 5
Router(config-sbc-sbe)# reject-threshold critical memory 15 9
Router(config-sbc-sbe)# end
Router# show sbc mySbc sbe call-stats reject-threshold

<table>
<thead>
<tr>
<th>Level</th>
<th>Memory Trigger</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>minor</td>
<td>&lt; 30 percent</td>
<td>0 in 10 calls dropped</td>
</tr>
<tr>
<td>major</td>
<td>&lt; 20 percent</td>
<td>5 in 10 calls dropped</td>
</tr>
<tr>
<td>critical</td>
<td>&lt; 15 percent</td>
<td>9 in 10 calls dropped</td>
</tr>
<tr>
<td>halt</td>
<td>&lt; 10 percent</td>
<td>10 in 10 calls dropped</td>
</tr>
</tbody>
</table>

Current level: NORMAL
Total calls rejected due to low memory threshold: 0

Image Upgrade Procedure for Cisco Unified Border Element
(SP Edition)

The following procedures describe how to perform an image upgrade.

Step 1  Copy the Cisco Unified Border Element (SP Edition) image from the tftp location onto your hard disk:

Step 2  Check if the node has two RP cards using the `show platform` command.

If the node has two RP cards, copy the image to the standby card using the following command:

`Router# copy harddisk:asr1000rp1new_image.bin stby-harddisk:asr1000rp1new_image.bin`

Step 3  Do a no boot system of the existing image on the Active RP using the following command:
Router(config)# no boot system harddisk:asr1000rp1old_image.bin

Step 4 Start the upgrade using the following command:
RTP-ASR-1(config)# boot system harddisk:asr1000rp1<new_image>.bin

Step 5 Do a `show run` to check if the changes are reflected.

Step 6 Reload the node using the `reload` command:
Router# reload

System configuration has been modified. Save? [yes/no]: y
Building configuration...
[OK]
Proceed with reload? [confirm] y

Step 7 To verify that the new image is loaded after the “reload,” use the `show version` command.

Step 8 After the upgrade, check that all the cards have come up in the Active state by using the `show platform` command.
Media Address Pools

You can configure Cisco Unified Border Element (SP Edition) with a single media address or a range of media addresses. In addition you can define one or more permissible port ranges for the configured addresses. This feature allows the administrator to configure or restrict the data border element (DBE) address by address pool with or without port range, and define class of service (CoS) affinity for each port range.

Note

For Cisco IOS XE Release 2.4, this feature is supported in both the unified and distributed models.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).

For a complete description of the commands used in this chapter, refer to the Cisco Unified Border Element (SP Edition) Command Reference: Unified Model at:


For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

Feature History for Media Address Pools

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 2.1</td>
<td>This feature was introduced on the Cisco IOS XR.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 2.4</td>
<td>Added support for SBC unified model.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 3.2S</td>
<td>Added support for media address pool selection using port range tags.</td>
</tr>
</tbody>
</table>

Contents

This chapter contains the following sections:

- Prerequisites—Implementing Media Address Pools, page 4-2
- Restrictions for Configuring Media Address Pools, page 4-2
- Media Address Pools, page 4-3
- Configuring Media Address Pools, page 4-3
Prerequisites—Implementing Media Address Pools

The following prerequisites are required to implement media address pools:

• Before implementing media address pools, you must create a static route.

  
  **Note** Creating a static route will fail if the remote peer is on the same VLAN as the interface VLAN of the media address.

• Before implementing media address pools, Cisco Unified Border Element (SP Edition) must already be configured.

Restrictions for Configuring Media Address Pools

The restrictions for configuring media address pools are:

• The ending address must be numerically higher than the starting address.
• The minimum port must be numerically lower than the maximum port.
• Port ranges may not overlap.
• Address ranges may not overlap.
• Address ranges and single addresses may not overlap.
• Where a range of addresses are defined in a single command, they will share any port ranges assigned. If there is a requirement to have different port ranges for different media addresses, then the addresses must be configured separately.
• Media addresses and port ranges may only be deleted before the DBE is activated. After DBE activation, the DBE must be deactivated in order to delete addresses and port ranges.
• After you configure media addresses and pools of addresses, you cannot delete them unless you delete the DBE.
• The port range tag is supported by only the signaling border element (SBE), and not the DBE.
• The media address and the signaling address should not be identical. If the media address and the signaling address are identical, and the Cisco ASR 1000 Series Router selects an ephemeral port to send out signaling packets, the port may overlap with the port range of the media address. As a result, the signaling packets do not get punted up to the RP, and get dropped by the media packet filter. This may result in events such as incomplete TCP handshakes during the second leg of a call through the SBC.
• The media address of the SBC must be unique, which means that:
  – It is not used by any features on the Cisco ASR 1000 Series Router other than sending and receiving call media.
  – It is not used by SBC call signaling.
Media Address Pools

If you do not specify a port range, all possible VoIP port numbers are valid. The full VoIP port range extends from 16384 to 32767 inclusive.

You can define a CoS affinity for each port range. The set of CoS is consistent with those used for Quality of Service (QoS) packet marking, and consists of voice and video. If you do not define an associated CoS affinity, then the affinity is for all call types.

You can modify the extent of the existing port ranges or the class of service (CoS) affinities of the existing port ranges, or delete an existing port range. Note that the configuration changes do not apply to the existing calls, but to the calls being set up after the configuration is committed.

From Cisco IOS Release 3.2S, support for selecting the media address pools using the port range tags has been added. A port range tag is a user-configured string that can be applied to a call in the Call Admission Control (CAC) policy in the SBC. A user can match the normal subset of call attributes when configuring a policy that applies a port range tag to a call, as with all the CAC policy fields. Similarly, tags can be added during the port range configurations on media addresses or media address pools.

When a call arrives at the SBC, it is passed to CAC as part of call setup. If a configured CAC policy matches the call, the policy assigns the port range tag to the call, after which the value is passed to the media component.

When selecting a local media address and port for a call, the SBC selects a port from a port range that can meet the following characteristics, which are applied in the order specified:

1. The media address range is in the requested VPN.
2. The media address range has an IP realm that matches the request for the media stream, if a media stream has been requested.
3. The port-range either has the same CoS configured as requested for the media stream, or has the "Any" CoS configured.
4. If the media stream has a port range tag specified, the port-range must have an identical port range tag configured. However, if the media stream does not have a port range tag specified, the port-range must have the default zero-length port range tag configured on it.

Configuring Media Address Pools

This section contains the steps for configuring media address pools.

SUMMARY STEPS

1. configure terminal
2. sbc service-name
3. media-address {ipv4 | ipv6} {addr} [nat-mode twice-nat | vrf vrf-name | managed-by {dbe | mgc}]
   or
   media-address pool {ipv4 | ipv6} {start-addr} {end-addr} [nat-mode twice-nat | vrf vrf-name | managed-by {dbe | mgc}]
4. port-range min-port max-port [any | voice | video | signaling | fax | tag tag-string]
5. end
6. show sbc service-name sbe addresses
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enables the global configuration mode.</td>
</tr>
</tbody>
</table>
| **Example:**  
Router# configure terminal | |
| **Step 2** sbc service-name | Creates the SBC service on the Cisco Unified Border Element (SP Edition) and enters the SBC configuration mode.  
Use the *service-name* argument to define the name of the SBC. |
| **Example:**  
Router(config)# sbc MySBC | |
| **Step 3** media-address {ipv4 | ipv6} {addr} [nat-mode twice-nat | vrf vrf-name | managed-by {dbe | mgc}] | Adds an IPv4 or IPv6 address to the set of addresses that can be used by the DBE as a local media address.  
- *addr*—Local IPv4 or IPv6 address on an SBC interface that can be used for media arriving on the DBE.  
- *nat-mode twice-nat*—(Optional) Allows local addresses to be reserved for Twice-NAT pinholes.  
- *vrf vrf-name*—(Optional) Specifies that the IP address is associated with a specific VPN routing and forwarding (VRF) instance. If the VRF is not specified, the address is assumed to be an address on the global VPN.  
- *managed-by*—(Optional) Specifies whether the DBE or the MGC is allowed to select these addresses as local addresses for flows.  
- *dbe*—(Optional) Specifies that only the DBE is allowed to select these addresses as local addresses for flows.  
- *mgc*—(Optional) Specifies that only the media gateway controller (MGC) is allowed to select these addresses as local addresses for flows. |
| **Example:**  
Router(config-sbc)# media-address ipv4 10.10.10.1 | |
Chapter 4  Media Address Pools

Configuring Media Address Pools

media-address pool {ipv4 | ipv6} {start-addr} {end-addr} {nat-mode twice-nat | vrf vrf-name | managed-by {dbe | mgc}}

Example:
Router(config-sbc)# media-address pool ipv4 10.10.10.1 10.10.10.20

Step 4  port-range min-port max-port [any | voice | video | signaling | fax | tag tag-string]

Example:
Router(config-sbc-media-address-pool)# port-range 16384 30000 video

Step 5  end

Example:
Router(config-sbc)# end

Step 6  show sbc sbe addresses

Example:
Router# show sbc dmsbc-node9 sbe addresses

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>media-address pool {ipv4</td>
<td>ipv6} {start-addr} {end-addr} {nat-mode twice-nat</td>
</tr>
<tr>
<td>start-addr—Starting IPv4 and IPv6 media address in a range of addresses.</td>
<td></td>
</tr>
<tr>
<td>end-addr—Ending IPv4 and IPv6 media address in a range of addresses. The ending address must be numerically greater than the starting address.</td>
<td></td>
</tr>
<tr>
<td>nat-mode twice-nat—(Optional) Allows local addresses to be reserved for Twice-NAT pinholes.</td>
<td></td>
</tr>
<tr>
<td>vrf vrf-name—(Optional) Specifies that the IP addresses are associated with a specific VRF instance. If the VRF instance is not specified, the address is assumed to be an address on the global VPN.</td>
<td></td>
</tr>
<tr>
<td>managed-by—(Optional) Specifies whether the DBE or the MGC is allowed to select these addresses as local addresses for flows.</td>
<td></td>
</tr>
<tr>
<td>dbe—(Optional) Specifies that only the DBE is allowed to select these addresses as local addresses for flows.</td>
<td></td>
</tr>
<tr>
<td>mgc—(Optional) Specifies that only the MGC is allowed to select these addresses as local addresses for flows.</td>
<td></td>
</tr>
</tbody>
</table>

| Step 4  port-range min-port max-port [any | voice | video | signaling | fax | tag tag-string] | Creates a pool of sequential IPv4 media addresses that can be used by the SBC as local media addresses, and enters the SBC media address pool configuration mode. |
| In the SBC media address pool configuration mode, the CoS for the port range is video. | |

| Step 5  end | Returns to the Privileged EXEC mode. |

| Step 6  show sbc sbe addresses | Lists the addresses configured on the SBEs. |
Note

There is a known issue for the **media-address** command. If a secondary IP address under an interface SBC is configured as a media-address, when you use the `no` form of the **media-address** command to remove that media-address, the corresponding secondary IP address under that interface SBC will be removed as well. Furthermore, if that secondary IP address is configured under some interface SBC both on Active and Standby (in B2B redundancy), removing that media-address will also remove that secondary IP address on Standby. For behaviors about IPv6 address under interface SBC are the same as that of secondary IPv4 address under interface SBC.

---

### Configuring the Port Range Tag for the CAC Policy

This section contains the steps to configure the port range tag for applying to a call in the CAC policy in the SBC.

**Note**

The **caller** and **callee** commands have been used in this procedure. In some scenarios, the **branch** command can be used as an alternative to the **caller** and **callee** command pair. The **branch** command has been introduced in Release 3.5.0. See the Help>Configuring Directed Nonlimiting CAC Policies? section on page 7-37 for information about this command.

---

### SUMMARY STEPS

1. configure terminal
2. sbc service-name
3. sbe
4. cac-policy-set policy-set-id
5. cac-table table-name
6. table-type {policy-set | limit {list of limit tables}}
7. entry entry-id
8. cac-scope {list of scope options}
9. caller port-range-tag {adj-name | none | string tag-string}
10. callee port-range-tag {adj-name | none | string tag-string}
11. action [next-table goto-table-name | cac-complete]
12. exit
13. exit
14. complete
15. exit
16. cac-policy-set global policy-set-id
17. end
18. show sbc sbc-name sbc cac-policy-set id table name entry id
<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><code>configure terminal</code></td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td>2</td>
<td><code>sbc service-name</code></td>
<td>Enters the SBC service mode.</td>
</tr>
<tr>
<td></td>
<td>Example: <code>Router# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Example: <code>Router(config)# sbc mysbc</code></td>
<td>Use the <code>service-name</code> argument to define the name of the service.</td>
</tr>
<tr>
<td>3</td>
<td><code>sbe</code></td>
<td>Enters the SBE entity mode within an SBC service.</td>
</tr>
<tr>
<td></td>
<td>Example: <code>Router(config-sbc)# sbe</code></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td><code>cac-policy-set policy-set-id</code></td>
<td>Enters the CAC policy set configuration mode within an SBE entity, creating a new policy set, if necessary.</td>
</tr>
<tr>
<td></td>
<td>Example: <code>Router(config-sbc-sbe)# cac-policy-set 1</code></td>
<td><code>policy-set-id</code>—The call policy set number that can range from 1 to 2147483647.</td>
</tr>
<tr>
<td>5</td>
<td><code>cac-table table-name</code></td>
<td>Enters the CAC table mode for configuration of an admission control table (creating one if necessary) within the context of an SBE policy set.</td>
</tr>
<tr>
<td></td>
<td>Example: <code>Router(config-sbc-sbe-cacpolicy)# cac-table StandardListByAccount</code></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>`table-type {policy-set</td>
<td>limit {list of limit tables}}`</td>
</tr>
<tr>
<td></td>
<td>Example: <code>Router(config-sbc-sbe-cacpolicy-cactable)# table-type policy-set</code></td>
<td>When the <code>policy-set</code> keyword is specified, use the <code>cac-scoped</code> command to configure the scope in each entry at which limits are applied in a CAC Policy Set table.</td>
</tr>
<tr>
<td></td>
<td>Example: <code>Router(config-sbc-sbe-cacpolicy-cactable-cac polítaction)cactable)# table-type policy-set</code></td>
<td><strong>Note</strong> In Policy Set tables, the event, call, or message is applied to all the entries.</td>
</tr>
<tr>
<td>7</td>
<td><code>entry entry-id</code></td>
<td>Enters the CAC table entry mode to create or modify an entry in an admission control table.</td>
</tr>
<tr>
<td></td>
<td>Example: <code>Router(config-sbc-sbe-cacpolicy-cactable)# entry 1</code></td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

**Step 8**

```
cac-scope (list of scope options)
```

### Purpose

Configures the scope within each entry at which limits are applied in a policy set table.

- **list of scope options**—Specifies one of the following strings used to match events:
  - `account`—Events that are from the same account.
  - `adjacency`—Events that are from the same adjacency.
  - `adj-group`—Events that are from members of the same adjacency group.
  - `call`—Scope limits are per single call.
  - `category`—Events that have same category.
  - `dst-account`—Events that are sent to the same account.
  - `dst-adj-group`—Events that are sent to the same adjacency group.
  - `dst-adjacency`—Events that are sent to the same adjacency.
  - `dst-number`—Events that have same destination.
  - `global`—Scope limits are global
  - `src-account`—Events that are from the same account.
  - `src-adj-group`—Events that are from the same adjacency group.
  - `src-adjacency`—Events that are from the same adjacency.
  - `src-number`—Events that have the same source number.
  - `sub-category`—The limits specified in this scope apply to all events sent to or received from members of the same subscriber category.
  - `sub-category-pfx`—The limits specified in this scope apply to all events sent to or received from members of the same subscriber category prefix.
  - `subscriber`—The limits specified in this scope apply to all events sent to or received from individual subscribers (a device that is registered with a Registrar server)

### Example:

```
Router(config-sbc-sbe-cacpolicy-cactable-entry)
 Router(config-sbc-sbe-cacpolicy-cactable-entry)
# cac-scope category
```
## Command or Action

| Step 9 | caller port-range-tag {adj-name | none | string tag-string} |
|--------|---------------------------------------------------------------|
| Example: | Router(config-sbc-sbe-cacpolicy-cactable-entry) | # caller port-range-tag adj-name |

Configures the port range tag for a caller. This tag is used when selecting the media address and port.

- **adj-name**—Uses the source adjacency name as a port range tag.
- **none**—Prompts the SBC to not use a port range tag for calls matching the CAC entry, and removes previously found strings, if any.
- **string tag-string**—Specifies the explicit port range tag string.

| Step 10 | callee port-range-tag {adj-name | none | string tag-string} |
|---------|---------------------------------------------------------------|
| Example: | Router(config-sbc-sbe-cacpolicy-cactable-entry) | # callee port-range-tag string GenericCorePortRange |

Configures the port range tag for a callee. This tag is used when selecting the media address and port.

- **adj-name**—Uses the destination adjacency name as a port range tag.
- **none**—Prompts the SBC to not use a port range tag for calls matching the CAC entry, and removes previously found strings, if any.
- **string tag-string**—Specifies the explicit port range tag string.

| Step 11 | action [next-table goto-table-name | cac-complete] |
|---------|---------------------------------------------------------------|
| Example: | Router(config-sbc-sbe-cacpolicy-cactable-entry) | # action cac-complete |

Configures the action to perform after this entry in an admission control table. Possible actions are:

- Identify the next CAC table to process using the **next-table** keyword and the **goto-table-name argument**.
- Stop processing for this scope using the **cac-complete** keyword.

<table>
<thead>
<tr>
<th>Step 12</th>
<th>exit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-cacpolicy-cactable-entry)</td>
</tr>
</tbody>
</table>

Exits the entry mode, and enters the CAC table mode.

<table>
<thead>
<tr>
<th>Step 13</th>
<th>exit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-cacpolicy-cactable-entry)</td>
</tr>
</tbody>
</table>

Exits the CAC table mode, and enters the CAC policy mode.

<table>
<thead>
<tr>
<th>Step 14</th>
<th>complete</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-cacpolicy)</td>
</tr>
</tbody>
</table>

Completes the CAC policy set after you commit the entire set.

<table>
<thead>
<tr>
<th>Step 15</th>
<th>exit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-cacpolicy)</td>
</tr>
</tbody>
</table>

Exits the CAC policy mode, and enters the SBE configuration mode.
This section provides sample configurations for media address pools. The following example shows the creation of a static route for the media pool address.

At the Route Processor (RP):

```
Router(config)# ip route 87.87.29.8 255.255.255.248 87.87.29.100
```

The following example creates a pool of IPv4 media addresses that can be used by the DBE as local media addresses:

```
Router(config)# sbc test dbe
Router(config-sbc-dbe)# media-address pool ipv4 87.87.29.8 87.87.29.15
```

The following sample script adds a single address (10.10.10.1), and two ranges of addresses (10.10.11.1 through 10.10.11.20 and 10.10.11.21 through 10.10.11.30) to the media address pool.

Two port ranges are configured on the single address. The first port range is for voice traffic, and runs from port 16384 to 20000 inclusively. The second one is for video traffic, and runs from port 20001 to 65535 inclusively.

The first range of addresses also has two similar port ranges configured that apply to all ten addresses within the range. The second range of addresses has a single port range defined, and no service class associated with it.

```
Router(config)# sbc test dbe
Router(config-sbc-dbe)# media-address pool ipv4 10.10.10.1
Router(config-sbc-dbe-media-address pool)# port-range 16384 20000 voice
Router(config-sbc-dbe-media-address pool)# exit

Router(config-sbc-dbe)# media-address pool ipv4 10.10.10.1
Router(config-sbc-dbe-media-address pool)# port-range 20001 65535 video
Router(config-sbc-dbe-media-address pool)# exit
```

---

### Configuring Media Address Pools: Example

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 16</td>
<td><strong>cac-policy-set global policy-set-id</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>Router(config-sbc-dbe)# cac-policy-set global 23</strong></td>
</tr>
<tr>
<td></td>
<td>• <strong>policy-set-id</strong>—The call policy set number, ranging from 1 to 2147483647. The policy set must be in a complete state before it can be assigned as the default policy.</td>
</tr>
<tr>
<td>Step 17</td>
<td><strong>end</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>Router(config-sbc-dbe-cacpolicy-cactable)# end</strong></td>
</tr>
<tr>
<td>Step 18</td>
<td><strong>show sbc sbc-name sbc cac-policy-set id table name entry id</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>Router# show sbc MySBC sbc cac-policy-set 1 table StandardListByAccount entry 1</strong></td>
</tr>
</tbody>
</table>
The following example shows how to add an IPv4 address to the set of addresses that can be used by the SBE as a local media address, and how to configure a port range tag:

```
sbc MySBC
  media-address ipv4 10.33.33.1
  port-range 2000 4000 voice tag GoldCustomerA
  port-range 4001 6000 video tag HighBwCustomer
  port-range 10000 12005 tag Adjacency_IMS_Core

  no port-range 10000 12005 tag
```

The following example shows how to create a pool of IPv6 media addresses that can be used by the SBE as local media addresses, and how to configure a port range tag:

```
sbc MySBC
  media-address pool ipv6 CAFE:1234:1234:1234::0001 CAFE:1234:1234:1234::0012
  port-range 2000 4000 voice tag LowBW@CustomerA
  port-range 4001 6000 signaling
  port-range 10000 12005 fax tag FaxGWAdjacency23
```

### Configuring a Port Range Tag for the CAC Policy: Example

This section provides a sample configuration of a port range tag for applying to a call in a CAC policy set in the SBC:

```
sbc MySBC
  sbe
    cac-policy-set 1
      cac-table Table1
        table-type policy-set instigate
        .
        .
        .
        entry 1
          cac-scope global
          caller port-range-tag adj-name
callee port-range-tag adj-name
          action next-table Table2
        .
        .
      cac-table Table2
        table-type limit account
        entry 1
          match-value GoldAccount
caller port-range-tag string LargeBWPorts
callee port-range-tag none
```
Implementing Multi-VRF on Cisco Unified Border Element (SP Edition)

Cisco Unified Border Element (SP Edition) provides support for multi-VRF (VPN routing and forwarding) on customer edge (CE) devices. This feature provides the capability of suppressing provider edge (PE) checks to prevent loops when the PE is performing a mutual redistribution of packets.

VRF is only supported in DBE media address and SBE AAA/H248 control address; DBE H248 control address does not support VRF.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).

For a complete description of the commands used in this chapter, refer to the Cisco Unified Border Element (SP Edition) Command Reference: Unified Model at:

For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

---

Note

For Cisco IOS XE Release 2.4, this feature is supported in both the unified and distributed model.

Feature History for Implementing Multi-VRF on Cisco Unified Border Element (SP Edition)

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 2.4</td>
<td>This feature was introduced on the Cisco ASR 1000 Series Routers.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 3.2S</td>
<td>SBC Voice traffic support over tunnel-interface (GRE, IPSec, MPLS, TE tunnel, BBA) was introduced on the Cisco ASR 1000 Series Routers.</td>
</tr>
</tbody>
</table>

Contents

This module contains the following sections:

- Prerequisites—Implementing Multi-VRF, page 5-2
- Information About Implementing Multi-VRF, page 5-2
- Implementing Multi-VRF, page 5-3
- Configuration Examples for Implementing Multi-VRF, page 5-7
- Supporting the SBC Voice Traffic over Tunnel Interfaces, page 5-13
Prerequisites—Implementing Multi-VRF

The following prerequisite is required to implement multi-VRF on Cisco Unified Border Element (SP Edition):

- Before implementing multi-VRF, Cisco Unified Border Element (SP Edition) must already be configured.

Information About Implementing Multi-VRF

Cisco Unified Border Element (SP Edition) support for multi-VRF on customer edge (CE) devices, such as customer premises routers, provides the capability of suppressing PE checks that are needed to prevent loops when the PE is performing a mutual redistribution of packets. Multi-VRF allows for the use of only one router to accomplish the tasks that multiple routers usually perform. It runs on a network without the requirement of MPLS and BGP installed.

When VRF is used on a router that is not a PE, the checks can be turned off to allow for correct population of the VRF routing table with routes to IP prefixes. Multi-VRF is also important because virtual private network (VPN) functionality is not completely supported on low-end systems. Multi-VRF provides logical separation of routing instances (and by the implication address space) within one router. The following summarizes the features of multi-VRF:

- Allows a single physical router to be split into multiple virtual routers, where each router contains its own set of interfaces, routing table, and forwarding table. Cisco Unified Border Element (SP Edition) supports multiple (overlapping and independent) routing tables (addressing) per customer. Virtual routing contexts are used to separate routing domains within a single router.
- Multi-VRF can be used where multiple routers are required but only one is available.
- When using multi-VRF, the domain name server (DNS) queries are per VRF.
- One physical interface can belong to multiple virtual routers through the use of subinterfaces (Frame Relay, ATM, VLANs).
- BGP and MPLS are not used.
- No connectivity is provided between VRFs (would require using BGP for internal exporting and importing between VRFs).
- When a call is placed between two endpoints in the same VPN site, Cisco Unified Border Element (SP Edition) can route the media directly between them, to reduce network utilization.
- Multi-VRF on Cisco Unified Border Element (SP Edition) provides optimization where both endpoints are on the same VPN by turning media bypass on.
- When a VRF is removed from a SBC interface that is in use by an activated SBC, the IP addresses are not removed automatically by the SBC. The user has to manually remove the IP addresses when the SBC is deactivated.

For Cisco IOS XE Release 2.4, by default, all adjacencies on the same VPN have media bypass turned on. Media bypass can be turned off by using the `media-bypass-forbid` command (this command is implemented for CAC policies only).

Note

The `vrf name` under the adjacency must match the context name.
Implementing Multi-VRF

Implementing multi-VRF is described in the following sections:

- Associating a SIP Adjacency with a VRF, page 5-3
- Configuring DBE with VRF—Distributed Model Only, page 5-5

Associating a SIP Adjacency with a VRF

This task associates a SIP adjacency with a VPN.

**Note**

When an adjacency is assigned to a particular VRF, all the addresses relating to the adjacencies, such as signalling-address and remote-address, must also be routable within the VRF.

**SUMMARY STEPS**

1. `adjacency sip adjacency-name`
2. `vrf vrf_name`
3. `signaling-address ipv4 local_signaling_IP_address`
4. `signaling-port port_num`
5. `remote-address ipv4 local_signaling_IP_address/prefix`
6. `local-id host name`
7. `signaling-peer peer_address`
8. `signaling-peer-port port_num`
9. `account account-name`
10. `media-bypass (optional)`
11. `media-bypass-forbid`
12. `attach`
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> adjacency sip adjacency-name</td>
<td>Enters the mode of an SBE SIP adjacency.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe)# adjacency sip sip_vrf1</td>
<td>Use the <strong>adjacency-name</strong> argument to define the name of the service.</td>
</tr>
<tr>
<td><strong>Step 2</strong> vrf vrf_name</td>
<td>Ties a SIP adjacency to a specific VPN.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-adj-sip)# vrf my_vrf1</td>
<td>The <strong>vrf name</strong> under the adjacency must match the context name.</td>
</tr>
<tr>
<td><strong>Step 3</strong> signaling-address ipv4 ipv4_IP_address</td>
<td>Specifies the local IPv4 signaling address of the SIP adjacency.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-adj-sip)# signaling-address ipv4 88.88.88.88</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> signaling-port port_num</td>
<td>Specifies the local signaling port of the SIP adjacency.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-adj-sip)# signaling-port 5060</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> remote-address ipv4 remote_IP_address/prefix</td>
<td>Restricts the set of remote signaling peers contacted over the adjacency to those with the given IP address prefix.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-adj-sip)# remote-address ipv4 10.10.101.4 255.255.255.255</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> local-id host address</td>
<td>Configures the local identity name on a SIP adjacency.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-adj-sip)# local-id host 88.88.101.11</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> signaling-peer peer_address</td>
<td>Specifies the remote signaling peer for the SIP adjacency to use.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-adj-sip)# signaling-peer 10.10.101.4</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> signaling-peer-port port_num</td>
<td>Specifies the remote signaling-peer port for the SIP adjacency to use.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-adj-sip)# signaling-peer-port 5060</td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong> account account_name</td>
<td>Defines the SIP adjacency as belonging to an account on an SBE.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-adj-sip)# account sip-vrf1</td>
<td></td>
</tr>
</tbody>
</table>
### Configuring DBE with VRF—Distributed Model Only

This task configures DBE with VRF in the distributed model.

**SUMMARY STEPS**

1. configure
2. sbc sbc-name dbe
3. vdbe global
4. unexpected-source-alerting
5. local-port abcd
6. control-address h248 ipv4 A.B.C.D
7. controller h248 controller-index
8. remote-address ipv4 remote-address
9. remote-port [port-num]
10. transport [udp | tcp]
11. attach-controllers
12. media-address pool ipv4 A.B.C.D E.F.G.H vrf vrfname
13. media-timeout timeout
14. overload-time-threshold time
15. deactivation-mode
16. activate

### Command or Action

<table>
<thead>
<tr>
<th>Step 10</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.</td>
<td>media-bypass</td>
<td>(Optional) Configures the adjacency to allow media traffic to bypass the DBE. This command is optional and only works on one adjacency.</td>
</tr>
</tbody>
</table>

**Example:**

```
Router(config-sbc-sbe-adj-sip)# media-bypass
```

<table>
<thead>
<tr>
<th>Step 11</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>11.</td>
<td>media-bypass-forbid</td>
<td>Configures the SIP adjacency to forbid media traffic to bypass the DBE. If this is not configured, media traffic for calls originating and terminating on this adjacency flows directly between the endpoints and does not pass through the DBE, as long as both adjacencies are on the same VPN.</td>
</tr>
</tbody>
</table>

**Example:**

```
Router(config-sbc-sbe-adj-sip)# media-bypass-forbid
```

<table>
<thead>
<tr>
<th>Step 12</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.</td>
<td>attach</td>
<td>Attaches the adjacency.</td>
</tr>
</tbody>
</table>

**Example:**

```
Router(config-sbc-sbe-adj-sip)# attach
```
# DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Accesses the configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc sbc-name dbec</td>
<td>Creates the DBE service on the SBC and enters into SBC-DBE configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# sbc mySbc</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> vdbe [global]</td>
<td>Enters into vDBE configuration submode.</td>
</tr>
<tr>
<td><strong>Note</strong> In the initial release only one vDBE (the global vDBE) is supported. The vbe name is not required. If specified, it must be global.</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-dbe)# vdbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> unexpected-source-alerting</td>
<td>Sets alerting for unexpected source addresses.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-dbe-vdbe-global)# unexpected-source-alerting</td>
<td>The no form of this command removes alerting for any unexpected source addresses that are received.</td>
</tr>
<tr>
<td><strong>Step 5</strong> local-port (abcd)</td>
<td>Configures a DBE to use a specific local port.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-dbe)# local-port 5090</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> control-address h248 ipv4 A.B.C.D</td>
<td>Configures a DBE to use a specific IPv4 H.248 control address.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-dbe)# control-address h248 ipv4 10.0.0.1</td>
<td>The control address cannot be in a VRF and must be routable in the global address table.</td>
</tr>
<tr>
<td><strong>Step 7</strong> controller h248 controller-index</td>
<td>Identifies the H.248 controller for the DBE and enters into Controller H.248 configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-dbe)# controller h248 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> remote-address ipv4 remote-address</td>
<td>Configures the IPv4 remote address of the H.248 controller.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-dbe-vdbe-h248)# remote-address ipv4 1.1.1.1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong> remote-port [port-num]</td>
<td>Defines the port to connect to on the SBE for an H.248 controller.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-dbe-h248)# remote-port 2094</td>
<td></td>
</tr>
</tbody>
</table>
### Configuration Examples for Implementing Multi-VRF

This section provides the following configuration examples:

- Configuring SBC Unified Model with VRF: Example, page 5-8
- Configuring Multi-VRF: Example, page 5-9
- Associating a SIP Adjacency with a VRF: Example, page 5-9
- Configuring DBE with Multi-VRF (Distributed Model Only): Example, page 5-11

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>transport udp</td>
<td>Configures a DBE to use User Datagram Protocol (UDP) for H.248 control signaling.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-dbe-h248)# transport udp</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>attach-controllers</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-dbe)# attach-controllers</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>media-address pool ipv4 A.B.C.D E.F.G.H vrf vrfname</td>
<td>Create a pool of sequential IPv4 media addresses for an IPv4 address associated with a specific VRF instance.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-dbe)# media-address pool ipv4 10.10.10.1 10.10.10.20 vrf my_vrf1</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
<td>The vrf name under the adjacency must match the context name.</td>
</tr>
<tr>
<td>13</td>
<td>media-timeout timeout</td>
<td>Sets the maximum time a DBE waits after receiving the last media packet on a call and before cleaning up the call resources.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-dbe)# media-timeout 10</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>overload-time-threshold time</td>
<td>Configures the threshold for media gateway (MG) overload control detection.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-dbe)# overload-time-threshold 400</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>deactivation-mode normal</td>
<td>Specifies that the DBE of an SBC signals a service change and terminates all calls upon deactivation of the DBE service.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-dbe)# deactivation-mode normal</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>activate</td>
<td>Initiates the SBC service.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-dbe)# activate</td>
<td></td>
</tr>
</tbody>
</table>
Configuring SBC Unified Model with VRF: Example

You can configure the Cisco ASR 1000 Series Router so that traffic is routed to the SBC adjacency address. This is achieved by creating a VRF instance on the router.

The following is an example, which uses VLAN trunks to get the traffic into the SBC. In this example, a VRF is created to route traffic from the 100.0.0.0/24 network to the 12.0.0.0/24 network, where the SIP signaling address and media address reside for a particular SBC connection.

The `interface sbc` command is needed, whenever a VRF is being used. You must have a secondary IP address defined if the media IP address is going to be different than the signaling address. However, in this case the secondary IP address is automatically added when the `media-address ipv4` command is used. It must not be manually entered.

```plaintext
vrf definition cust100side // Create a VRF instance

! address-family ipv4
exit-address-family
interface SBC100 // Create an interface in the VRF space
vrf forwarding cust100side
ip address 12.0.0.30 255.255.255.0 secondary // This contains the IP address for the
  // media, if different to the signaling
  // address. The line is not entered, but
  // appears automatically after the DBE
  // configuration is entered (see
  // 'media-address' CLI later.)

ip address 12.0.0.20 255.255.255.0 // This is the SIP adjacency address

interface GigabitEthernet0/1/0
  no ip address
  media-type sfp
  negotiation auto

interface GigabitEthernet0/1/0.100 // VLAN identifier 100 defined here
vrf forwarding cust100side
encapsulation dot1Q 100
ip address 100.0.0.1 255.255.255.0 // This IP is where the remote side or external
  // router can send traffic to, in order to get
  // to the internal 12.0.0.0/24 network

interface GigabitEthernet0/1/0.200 // Other VLANS that are being trunked.
vrf forwarding cust200side
encapsulation dot1Q 200
ip address 200.0.0.1 255.255.255.0

sbc ted
sbe
  adjacency sip adj_cust100
    vrf cust100side
      ...
      signaling-address 12.0.0.20 // This is the local address where call traffic
        // will get routed to/from
      remote-address ipv4 100.0.0.14 // This is an address for the remote side, where
        // traffic will be routed
      ...
      attach
      ...
  media-address ipv4 12.0.0.30 vrf cust100side // The media address is also on the
    // internal network. When the line
    // is entered, the interface SBC
    // will show a secondary address
    // containing this IP address.

activate
```
Configuring Multi-VRF: Example

This sample configuration shows how the Service Virtual Interface (SVI) and adjacencies are added to associate a VPN to them.

1. Configure the line card interface associated with vrf my_vrf1 on the route processor (RP).

   vrf definition my_vrf1
   rd 55:1111
   !
   address-family ipv4
   exit-address-family
   !

2. Configure the line card interface associated with vrf, my_vrf1, on the route processor.

   interface GigabitEthernet1/3
   description "Connected to CAT-3550-101 Fa 0/13 vlan919"
   ip address 10.122.3.3 255.255.255.0

   interface GigabitEthernet1/3.99
   encapsulation dot1q 99
   vrf forwarding my_vrf1
   ip address 10.122.3.3 255.255.255.0
   !

3. Configure the media address pools.

   media-address pool ipv4 88.88.101.12 88.88.101.15 vrf my_vrf1 activate

Associating a SIP Adjacency with a VRF: Example

This example configuration creates a SIP adjacency associated with a VPN.

   ip route 10.10.0.0 255.255.0.0 101.101.101.100 ip route 20.20.20.0 255.255.255.0 101.101.101.4

   domain default-domain

   sbc mysbc
   sbe
   adjacency sip 7200-1
     vrf my_vrf1
     inherit profile preset-core
     preferred-transport udp
     redirect-mode pass-through
     authentication nonce timeout 300
     signaling-address ipv4 101.101.101.3
     signaling-port 5061
     remote-address ipv4 0.0.0.0 0.0.0.0 signaling-peer 101.101.101.5 signaling-peer-port 5060 account sip-core attach
   adjacency sip 7200-2
     vrf my_vrf1
     inherit profile preset-access
     preferred-transport udp
     redirect-mode pass-through
     authentication nonce timeout 300
     signaling-address ipv4 101.101.101.3
signaling-port 5060
remote-address ipv4 0.0.0.0 0.0.0.0
signaling-peer 101.101.101.4
signaling-peer-port 5060
account sip-core
attach

adjacency sip 7200-3
vrf my_vrf1
nat force-on
inherit profile preset-core
preferred-transport udp
redirect-mode pass-through
authentication nonce timeout 300
signaling-address ipv4 101.101.101.3
signaling-port 5063
remote-address ipv4 0.0.0.0 0.0.0.0
signaling-peer 101.101.101.5
signaling-peer-port 5063
account sip-core
reg-min-expiry 3000
attach

sip inherit profile preset-standard-non-ims
retry-limit 3

call-policy-set 1
  first-call-routing-table invite-table
  first-reg-routing-table start-table
  rtg-src-adjacency-table invite-table
    entry 1
      action complete
dst-adjacency 7200-2
match-adjacency 7200-3
    entry 2
      action complete
dst-adjacency 7200-3
match-adjacency 7200-2
rtg-src-adjacency-table start-table
    entry 1
      action complete
dst-adjacency 7200-1
match-adjacency 7200-2
    entry 2
      action complete
dst-adjacency 7200-2
match-adjacency 7200-1
complete

active-call-policy-set 1

network-id 2

sip max-connections 2
sip timer
tcp-idle-timeout 120000
tls-idle-timeout 3600000
udp-response-linger-period 32000
udp-first-retransmit-interval 500
udp-max-retransmit-interval 4000
invite-timeout 180
blacklist
**Chapter 5 Implementing Multi-VRF on Cisco Unified Border Element (SP Edition)**

**Configuration Examples for Implementing Multi-VRF**

```

global

   redirect-limit 2
   deactivation-mode normal
   activate

media-address ipv4 101.101.101.160 vrf my_vrf1 port-range 11000 20000 any
location-id 0
media-timeout 30
deactivation-mode normal
activate
```

## Configuring DBE with Multi-VRF (Distributed Model Only): Example

To make use of Multi-VRF when Cisco Unified Border Element (SP Edition) is running in the distributed mode, both the configuration and the corresponding H.248 messages are required to be VRF-aware.

The following sample configuration creates media pool that is tied to a particular VRF. This media pool can only be used to assign media addresses for that particular VRF and can overlap with addresses from different VRFs or from the global address space.

```
vrf definition moon
   vpn id 22AA:33334411
!
interface SBC1
   ip vrf forwarding moon
   ip address 90.0.0.1 255.0.0.0
!
sbc global dbe
vdbe global
h248-version 3
h248-napt-package napt
local-port 2979
control-address h248 ipv4 200.50.1.9
controller h248 1
   remote-address ipv4 200.50.1.254
   remote-port 2979
attach-controllers
location-id 1
media-address ipv4 90.0.0.1 vrf moon
   port-range 10000 20000 any
activate
!
```
The H.248 configuration is specified in the H.248 package/Extended VPN Discrimination/ (EVPND). This package has two methods, GVPNID and VRF_NAME, of specifying to which VRF the media addresses belong. These parameters are mutually exclusive but they are independent on a per side basis. For example, side A may use the VRF_NAME method for specifying the VRF and side B may use the GVPNID method.

The VRF_NAME is a quoted ASCII string corresponding to the name of the VRF in the configuration. In the following example, the name would be "moon."

```
M {
    TS { SI = IV },
    ST = 1 {
        O { MO = IN,
            EVPND/VRF_NAME = "moon"
        },
        R {
            v=0
            c=IN IP4 3.0.0.3
            m=application 5000 udp 0
        },
        L {
            v=0
            c=IN IP4 $
            m=application $ udp 0
        }
    }
}
```

The GVPNID is the identification number for the VRF in RFC2685 format. This is specified in the configuration as follows:

```
vrf definition moon
    vpn id 22AA:33334411
```

The H.248 format is then specified as:

```
M {
    TS { SI = IV },
    ST = 1 {
        O { MO = IN,
            EVPND/GVPNID = 22AA33334411
        },
        R {
            v=0
            c=IN IP4 3.0.0.3
            m=application 5000 udp 0
        },
        L {
            v=0
            c=IN IP4 $
            m=application $ udp 0
        }
    }
}
```
Supporting the SBC Voice Traffic over Tunnel Interfaces

The Cisco IOS XE Release earlier than Cisco IOS XE Release 3.2S did not support the SBC traffic over the tunnel interfaces. The Cisco IOS XE Release 3.2S provides support to the SBC traffic over the tunnel interfaces (PPPoE, GRE, MPLS-TE, IPsec SVTI or DVTI, DMVPN). The following topology diagrams (Figure 5-1 and Figure 5-2) illustrate the broadband deployment scenario and tunnel interface scenarios in which the SBC voice traffic is supported over the tunnel interfaces:

**Figure 5-1** Broadband Deployment Topology Supporting the SBC Traffic

**Figure 5-2** IPSec Tunnel Deployment Topology Supporting the SBC Traffic
Supporting the SBC Voice Traffic over Tunnel Interfaces
Implementing Adjacencies on Cisco Unified Border Element (SP Edition)

Accounts and adjacencies are the key objects used to control signaling. An account represents a service relationship with a remote organization on the signaling border element (SBE), with which Cisco Unified Border Element (SP Edition) will interact. Within each account, the user defines one or more signaling adjacencies, which connect Cisco Unified Border Element (SP Edition) to devices within that organization. The account is used to:

- Define customer-specific admission control
- Define routing policy configurations
- Organize billing records

An adjacency represents a signaling relationship with a remote call agent. There is one adjacency defined per external call agent. The adjacency is used to define protocol-specific parameters as well as admission control and routing policy. Each adjacency belongs within an account.

Each incoming call is matched to an adjacency, and each outgoing call is routed out over a second adjacency. Adjacencies can also be associated with a media gateway location, so that the most appropriate virtual data border element (vDBE) can be selected for a given call leg. Typically, an Cisco Unified Border Element (SP Edition) has at least one account representing the internal network.

You can assign each adjacency to an adjacency group, so you can enable and disable features per interface. For example, you can turn off high bandwidth features on all adjacencies to customers on a known low-bandwidth link.

This chapter also discusses the SIP Over Transport Layer Security (TLS) feature, an encryption feature that provides a secure, encrypted transport to carry all SIP messages from the caller to the callee’s domain.

---

**Note**

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).

**Note**

For Cisco IOS XE Release 2.4, this feature is supported in the unified model only.
For a complete description of the commands used in this chapter, refer to the *Cisco Unified Border Element (SP Edition) Command Reference: Unified Model* at:


For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

**Feature History for Implementing Adjacencies and SIP Over TLS**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 2.4</td>
<td>This feature and SIP over TLS were introduced on the Cisco IOS XR</td>
</tr>
<tr>
<td></td>
<td>along with support for the unified model.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 2.6</td>
<td>The following features were added:</td>
</tr>
<tr>
<td></td>
<td>• Configurable Mutual TLS Authentication Per Interface.</td>
</tr>
<tr>
<td></td>
<td>• TLS Transport Parameter in Record Route Headers.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 3.1S</td>
<td>The Redundant Peer Addresses feature was added.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 3.2S</td>
<td>The SIP peer availability detection feature was added.</td>
</tr>
<tr>
<td></td>
<td>The Public Key Infrastructure (PKI) High Availability (HA) support was</td>
</tr>
<tr>
<td></td>
<td>added.</td>
</tr>
</tbody>
</table>

**Contents**

This module contains the following sections:

- Prerequisites for Implementing Adjacencies, page 6-2
- Restrictions, page 6-3
- Information About Implementing Adjacencies, page 6-3
- How to Implement Adjacencies, page 6-7
- Configuration Examples for Implementing Adjacencies, page 6-14
- SIP UAS Failure Detection, page 6-15
- SIP Outbound Flood Protection, page 6-18
- SIP Over TLS, page 6-20
- SIP Peer Availability Detection, page 6-32
- Redundant Peer Addresses, page 6-34

**Prerequisites for Implementing Adjacencies**

The following prerequisite is required to implement adjacencies:

- Before implementing adjacencies, Cisco Unified Border Element (SP Edition) must already be configured.
Restrictions

H.323 adjacencies are not supported in Cisco IOS XE Release 2.4 and earlier.

Information About Implementing Adjacencies

Adjacencies are used to enable call signaling between the SBE and other voice over IP (VoIP) devices. Cisco Unified Border Element (SP Edition) supports adjacencies in Session Initiation Protocol (SIP) network deployments.

In a SIP network, the devices might be user agents, proxies, softswitches, or back-to-back user agents (B2BUAs). When you configure a SIP adjacency, the SBE functions as a B2BUA within the SIP network.

Adjacencies can represent both trunking and subscriber signaling relationships. The network topology and configuration of an adjacency determine its role.

The adjacency accepts packets from either the UDP or TCP socket specified in the signaling port configuration line. For SIP, the default is port 5060. When sending packets out the adjacency, the transport used is specified using the `preferred-transport [tcp | udp]` command. The default is to use UDP. Note that there is no dependency between the input and output adjacencies. It is valid to have one adjacency use TCP for the signaling and the other use UDP.

Further overview details about implementing adjacencies are described in the following sections:

- Properties Common to SIP Adjacencies
- About SIP Adjacencies in the Deployment
- How Adjacencies Affect Media Routing

Properties Common to SIP Adjacencies

The following properties are common to SIP adjacencies:

- Adjacencies are known by name. The name makes it easy for a Cisco Unified Border Element (SP Edition) policy to reference the adjacency.
- An adjacency has a local address and port for incoming call setup.
- An adjacency has a peer address and port. This is the point of contact for outgoing calls. In the SIP case, this is only true if the "force-signaling-peer" option is set for that adjacency.
- An adjacency forms the output of a routing policy decision. In other words, the routing phase for a call results in selection of an outgoing adjacency for that call. Normally, adjacency selection is done based on a destination telephone number prefix. However, two adjacencies can also be bridged together by using a source adjacency as a routing input.

About SIP Adjacencies in the Deployment

Figure 6-1 shows a simple SIP network where:

- SIP subscribers register with the SIP proxy, which acts as a single point of contact for all of them.
- The softswitch is a gateway between the SIP network and the public switched telephone network (PSTN).
The softswitch routing policy assigns a particular phone prefix to each SIP proxy, allowing calls from the PSTN network to be routed through the proxy to a given subscriber. (In other deployments, subscribers may register directly with a softswitch without going through a proxy first.)

**Figure 6-1**  SIP Network

![SIP Network Diagram]

**Figure 6-2** shows placement of a Cisco Unified Border Element (SP Edition) in two possible positions within the SIP network, with the adjacencies noted. Each adjacency enables call setup to one or more neighboring devices, as follows:

- ADJ_SIP1A allows call setup between SBC1 and the softswitch.
- ADJ_SIP1B allows call setup between SBC1 and the proxy.
- ADJ_SIP2A allows call setup between SBC2 and the proxy.
- ADJ_SIP_SUBSCRIBERS allows call setup between SBC2 and the subscribers.

In the case of SBC2, SIP registrations are being routed through the SBC. Registrations received on ADJ_SIP_SUBSCRIBERS are being routed to the proxy over ADJ_SIP2A.

The key difference between subscriber and nonsubscriber adjacencies is that:

- Nonsubscriber adjacencies have a configured single point of contact, the peer address for the adjacency.
- Subscriber adjacencies do not have a single point of contact and are instead configured to accept registrations.

SIP registrations require a routing policy to determine which is the correct outgoing adjacency for a given registration. This works in a very similar way to a call routing policy. See the procedures described in the Implementing Cisco Unified Border Element (SP Edition) Policies module.
How Adjacencies Affect Media Routing

For a distributed Cisco Unified Border Element (SP Edition) deployment, each adjacency is configured with a *media location*. The media location is an ID used to select the data border elements (DBEs) suitable for relaying media traffic for calls set up over the adjacency.

If a call is routed out over the same or different adjacency, the media may bypass a DBE. The media bypass feature allows the media packets to bypass the Cisco Unified Border Element (SP Edition) to enable the endpoints to communicate directly to each other. Media packets flow directly without going through the DBE component of the SBC after the call signaling is done. Signaling packets still flow through the SBC as usual.

The configuration is set per adjacency, and allows media bypass across different adjacencies. Media-bypass configuration is enabled under adjacency configuration. Media bypass is useful when two endpoints are on the same subnet, but the DBE is located elsewhere on the network.

*Figure 6-3* and *Figure 6-4* illustrate how adjacency configuration controls media routing. In this example:

- Adjacency A connects to Peer1
- Adjacency B connects to Peer2a and 2b
- Adjacency C connects to Peer3
Adjacencies A and B are configured with media location 1. In other words, calls routed over them will use the same DBE (or set of DBEs) for media. Adjacency C is configured with media location 2.

**Figure 6-3 How Adjacency Configuration Controls Media Routing**

Now consider three calls: Peer1-Peer3, Peer1-Peer2a, and Peer2a-Peer2b. The media for these calls is routed as shown in **Figure 6-4**.

- The first call traverses two adjacencies with different media locations. Its media is relayed through two DBEs.
- The second call traverses two adjacencies with the same media location. Its media is relayed through a single DBE.
- The third call traverses a single adjacency with media bypass enabled. Its media is sent directly between the two peers without involving a DBE.

**Figure 6-4 Media Routing for Three Calls: Peer1-Peer3, Peer1-Peer2a, and Peer2a-Peer2b**
How to Implement Adjacencies

Adjacencies are the key objects used to control signaling. The user defines one or more signaling adjacencies, which connect the Cisco Unified Border Element (SP Edition) to devices within that organization. Each incoming call is matched to an adjacency, and each outgoing call is routed out over an adjacency. The adjacencies are then attached to the appropriate account. Adjacencies can be associated with a media gateway DBE location, so that the most appropriate DBE can be selected to route media for a given call leg.

Note

The default behavior for Cisco Unified Border Element (SP Edition) is to route INVITE requests to the device specified in the Request URI. If instead, the user wishes requests to be routed to the signaling peer, then 'force-next-hop' behavior should be enabled by configuring the force-signaling-peer command on the outbound adjacency.

The following sections describe implementing a SIP adjacency, depending on your implementation requirements:

- Configuring Force-Signaling-Peer Adjacency, page 6-7
- Configuring a SIP Adjacency, page 6-8
- Assigning SIP Adjacencies to Adjacency Groups, page 6-13

Configuring Force-Signaling-Peer Adjacency

This task configures a force-signaling-peer adjacency.

SUMMARY STEPS

1. configure terminal
2. sbc service-name
3. sbe
4. adjacency sip adjacency-name
5. force-signaling-peer
6. attach
7. exit
Chapter 6  Implementing Adjacencies on Cisco Unified Border Element (SP Edition)

How to Implement Adjacencies

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc service-name</td>
<td>Enters the mode of an SBC service. Use the service-name argument to define the name of the service.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# sbc umsbc-node10</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Enters the mode of an SBE entity within an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> adjacency sip adjacency-name</td>
<td>Enters the mode of an SBE SIP adjacency. Use the adjacency-name argument to define the name of the SIP adjacency.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe)# adjacency sip 2651XM-5</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> force-signaling-peer</td>
<td>Forces SIP messages to go to the configured signaling peer.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-adj-sip)# force-signaling-peer</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> attach</td>
<td>Attaches the adjacency.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-adj-sip)# attach</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> exit</td>
<td>Exits the sip mode to the sbe mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-adj-sip)# exit</td>
<td></td>
</tr>
</tbody>
</table>

Configuring a SIP Adjacency

You can only modify adjacencies when the adjacency is detached. Before modifying an adjacency, you can detach the adjacency first with the no attach command. The adjacency stays in the going down state when a call is active or when the ping enable feature is running. During this state, existing calls are not torn down and new calls are not accepted. The adjacency does not go to detached state until all calls have ended. An adjacency cannot be attached until the adjacency is in detached state.

If you wish to override the option to wait till active calls on the adjacency end, the adjacency can be detached immediately using the following commands:

- **no attach force abort**—Executes a forced detach, tearing down calls without signaling their end.
- **no attach force normal**—Executes a forced detach, tearing down calls gracefully.
To check the state of the adjacency, you can use the `show sbc sbe adjacencies` command.

---

**Caution**

Adjacencies can only be modified when the status is detached. Before modifying an adjacency, use the `no attach` command first.

---

**Note**

For User-to-Network Interface (UNI) registration support for a SIP inherit profile, you have the option of using the default value or a preset-access or a preset-core value. When using the default value for those adjacencies without specific per adjacency configuration, the `sip inherit profile preset-standard-non-ims` command in the SBE configuration mode (config-sbc-sbe) is applied to the adjacencies by default, and UNI registration support is enabled for this default configuration. When configuring a a preset-access or a preset-core value, use the `inherit profile preset-p-cscf-access` command on the adjacency facing subscribers and the `inherit profile preset-p-cscf-core` command on the adjacency facing the SIP proxy. If you use other combinations (for example, if both the inbound and outbound adjacencies are configured as preset-core, Cisco Unified Border Element (SP Edition) will not store the registration information, nor will it rewrite the Contact: header to make sure it’s on the signaling path of future messages.

This task configures two session initiation protocol (SIP) adjacencies. The first adjacency is configured for a gateway/endpoint. The second adjacency is configured with proxy/softswitch.

**SUMMARY STEPS**

1. `configure terminal`
2. `sbc service-name`
3. `sbe`
4. `sip inherit profile {preset-ibcf-ext-untrusted | preset-ibcf-external | preset-ibcf-internal | preset-p-cscf-access | preset-p-cscf-core | preset-standard-non-ims}`
5. `adjacency sip adjacency-name`
6. `signaling-address ipv4 ipv4_IP_address`
7. `signaling-port port_num`
8. `remote-address ipv4 ipv4_IP_address/prefix`
9. `signaling-peer peer_address`
10. `signaling-peer-port port_num`
11. `account account-name`
12. `registration rewrite-register`
13. `attach`
14. `exit`
15. `adjacency sip adjacency-name`
16. `inherit profile {preset-access | preset-core | preset-ibcf-ext-untrusted | preset-ibcf-external | preset-ibcf-internal | preset-p-cscf-access | preset-p-cscf-core | preset-peering | preset-standard-non-ims}`
17. `signaling-address ipv4 ipv4_IP_address`
18. `signaling-port port_num`
19. remote-address ipv4 \textit{ipv4\_IP\_address/prefix}
20. fast-register disable
21. signaling-peer \textit{peer\_name}
22. signaling-peer-port \textit{port\_num}
23. account \textit{account\_name}
24. registration target address \textit{host\_address}
25. registration target port \textit{port\_num}
26. attach
27. exit
28. end
29. show

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc \textit{service-name}</td>
<td>Enters the mode of an SBC service. Use the \textit{service-name} argument to define the name of the service.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Enters the mode of an SBE entity within an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# sbc mysbc</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> sip inherit profile (\textit{preset-ibcf-ext-untrusted}</td>
<td>Configures the global default inherit profile for all adjacencies.</td>
</tr>
<tr>
<td>\textit{preset-ibcf-external}</td>
<td></td>
</tr>
<tr>
<td>\textit{preset-ibcf-internal}</td>
<td></td>
</tr>
<tr>
<td>\textit{preset-p-cscf-access}</td>
<td></td>
</tr>
<tr>
<td>\textit{preset-p-cscf-core}</td>
<td></td>
</tr>
<tr>
<td>\textit{preset-standard-non-ims})</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc)# sip inherit profile preset-standard-non-ims</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> adjacency sip \textit{adjacency-name}</td>
<td>Enters the mode of an SBE SIP adjacency. Use the \textit{adjacency-name} argument to define the name of the service.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe)# adjacency sip sipGW</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> signaling-address ipv4 \textit{ipv4_IP_address}</td>
<td>Specifies the local IPv4 signaling address of the SIP adjacency.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-adj-sip)# signaling-address ipv4 88.88.141.3</td>
<td></td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
</tr>
<tr>
<td>------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td>7</td>
<td><strong>signaling-port</strong> <code>port_num</code></td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td>8</td>
<td><strong>remote-address</strong> <code>ipv4</code> <code>ipv4_IP_address/prefix</code></td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td>9</td>
<td><strong>signaling-peer</strong> <code>peer_address</code></td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td>10</td>
<td><strong>signaling-peer-port</strong> <code>port_num</code></td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td>11</td>
<td><strong>account</strong> <code>account_name</code></td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td>12</td>
<td><strong>registration rewrite-register</strong></td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td>13</td>
<td><strong>attach</strong></td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td>14</td>
<td><strong>exit</strong></td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td>15</td>
<td><strong>adjacency sip</strong> <code>adjacency-name</code></td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
</tbody>
</table>
### Chapter 6  Implementing Adjacencies on Cisco Unified Border Element (SP Edition)

#### How to Implement Adjacencies

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-adj-sip)# inherit profile preset-standard-non-ims</td>
<td></td>
</tr>
<tr>
<td>Step 17</td>
<td>signaling-address ipv4 ipv4_IP_address</td>
<td>Specifies the local IPv4 signaling address of the SIP adjacency.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-adj-sip)# signaling-address ipv4 88.88.141.11</td>
<td></td>
</tr>
<tr>
<td>Step 18</td>
<td>signaling-port port_num</td>
<td>Specifies the local signaling port of the SIP adjacency.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-adj-sip)# signaling-port 5060</td>
<td></td>
</tr>
<tr>
<td>Step 19</td>
<td>remote-address ipv4 ipv4_IP_address/prefix</td>
<td>Restricts the set of remote signaling peers contacted over the adjacency to those with the given IP address prefix.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-adj-sip)# remote-address ipv4 200.200.200.0/24</td>
<td></td>
</tr>
<tr>
<td>Step 20</td>
<td>fast-register disable</td>
<td>Disables fast register support on the SIP adjacency.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-adj-sip)# fast-register disable</td>
<td></td>
</tr>
<tr>
<td>Step 21</td>
<td>signaling-peer peer_address</td>
<td>Specifies the remote signaling peer for the SIP adjacency to use.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-adj-sip)# signaling-peer 200.200.200.98</td>
<td></td>
</tr>
<tr>
<td>Step 22</td>
<td>signaling-peer-port port_num</td>
<td>Specifies the remote signaling-peer port for the SIP adjacency to use.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-adj-sip)# signaling-peer-port 5060</td>
<td></td>
</tr>
<tr>
<td>Step 23</td>
<td>account account_name</td>
<td>Defines the SIP adjacency as belonging to an account on an SBE.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-adj-sip)# account COREvlan</td>
<td></td>
</tr>
</tbody>
</table>
### Assigning SIP Adjacencies to Adjacency Groups

Use the procedure in this section to assign an SIP adjacency to an adjacency group.

#### SUMMARY STEPS

1. `configure terminal`
2. `sbc service-name`
3. `sbe`
4. `adjacency sip adjacency-name`
5. `group adjacency-group-name`
6. `end`
7. `show`

---

#### Command or Action | Purpose
--- | ---
**Step 24**

`registration target address host_address` <br> **Example:**<br>Router(config-sbc-sbe-adj-sip)# registration target address 200.200.200.98

Sets the address to use if rewriting an outbound SIP REGISTER request.

**Step 25**

`registration target port port_num` <br> **Example:**<br>Router(config-sbc-sbe-adj-sip)# registration target port 5060

Sets the port to use if rewriting an outbound SIP REGISTER request.

**Step 26**

`attach` <br> **Example:**<br>Router(config-sbc-sbe-adj-sip)# attach

Attaches the adjacency.

**Step 27**

`exit` <br> **Example:**<br>Router(config-sbc-sbe-adj-sip)# exit

Exits `adj-sip` mode to `sbe` mode.

**Step 28**

`end` <br> **Example:**<br>Router(config-sbc-sbe-adj-sip)# end

Exits the `sbe` mode and returns to Privileged EXEC mode.

**Step 29**

`show` <br> **Example:**<br>Router(config-sbc-sbe)# show

Shows contents of configuration.
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong>  <code>configure terminal</code></td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong>  <code>sbc service-name</code></td>
<td>Enters the mode of an SBC service. Use the <code>service-name</code> argument to define the name of the service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config)# sbc mysbc</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong>  <code>sbe</code></td>
<td>Enters the mode of an SBE entity within an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-sbc)# sbe</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong>  <code>adjacency sip adjacency-name</code></td>
<td>Enters the mode of an SBE SIP adjacency. Use the <code>adjacency-name</code> argument to define the name of the service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-sbc-sbe)# adjacency sip sipGW</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong>  <code>group adjacency-group-name</code></td>
<td>Assigns the SIP adjacency to an adjacency group. Use the <code>adjacency-group-name</code> argument to define the group name.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-sbc-sbe-adj-sip)# group InternetEth0</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong>  <code>end</code></td>
<td>Exits <code>adj-sip</code> mode to <code>sbe</code> mode and returns to Privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-sbc-sbe-adj-sip)# end</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong>  <code>show</code></td>
<td>Shows contents of configuration.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router# show</code></td>
<td></td>
</tr>
</tbody>
</table>

### Configuration Examples for Implementing Adjacencies

This section provides the following configuration example:
- Configuring a SIP Adjacency: Example, page 6-15
Configuring a SIP Adjacency: Example

The following example configures two SIP adjacencies. The first adjacency is configured for a gateway/endpoint. The second adjacency is configured with proxy/softswitch.

1. Activate SBE, as follows:
   
   ```
   sbc sip-signal
   sbe
   activate
   exit
   ```

2. Activate DBE, as follows:
   
   ```
   sbc mySbc dbe
   media-address ipv4 88.88.141.2
   activate
   exit
   ```

3. Create the SIP adjacencies, as follows:
   
   ```
   sbc sip-signal
   sbe
   ```

4. Create the SIP adjacency for gateway/endpoint:
   
   ```
   adjacency sip sipGW
   signaling-address ipv4 88.88.141.3
   signaling-port 5060
   remote-address ipv4 10.10.121.0/24
   signaling-peer 10.10.121.10
   signaling-peer-port 5060
   account iosgw
   registration rewrite-register
   attach
   exit
   ```

5. Create the SIP adjacency for proxy/softswitch:
   
   ```
   adjacency sip sipPROXY
   signaling-address ipv4 88.88.141.11
   signaling-port 5060
   remote-address ipv4 200.200.200.0/24
   fast-register disable
   signaling-peer 200.200.200.98
   signaling-peer-port 5060
   account COREvlan
   registration target address 200.200.200.98
   registration target port 5060
   attach
   ```

SIP UAS Failure Detection

A User Agent Server (UAS) is a logical entity that generates a response to a SIP request. UAS failure detection is used to periodically monitor the state of a SIP network entity specified as the signaling peer on a SIP adjacency. SIP OPTIONS messages are sent to these network entities as a ping mechanism and a response from the device is expected. If a response is not received from the device, it is considered unreachable and removed from the routing calculations. Calls which cannot be routed through an alternate device are immediately responded to with a 604 Does Not Exist Anywhere message.
SIP UAS Failure Detection

Cisco Unified Border Element (SP Edition) by default acts as an UAS that responds to OPTION pings when OPTION pings are sent to it. SIP UAS Failure Detection enables Cisco Unified Border Element (SP Edition) to send a SIP OPTIONS message to the device specified in the SIP Adjacency Destination Address. If an acceptable response is received within the SIP transaction timeout period then the routing tables are updated and the device is considered routable.

A ping failure occurs when no acceptable response is received within the SIP transaction timeout period. If ping-fail-count failures occur, then the device is considered to be unreachable. The signaling peer is considered offline as far as routing is concerned. Cisco Unified Border Element (SP Edition) sends pings at the rate specified in the period.

**Note**

When the SBC has a TCP-based adjacency with OPTION ping enabled and that adjacency does not have a valid peer with which a TCP connection can be established, then that adjacency must be in the “no attach” state. This prevents the SBC from attempting to set up a TCP connection to a non-existent peer to send an OPTIONS ping message.

Use the procedure in this section to configure SIP UAS Failure Detection:

**SUMMARY STEPS**

1. configure terminal
2. sbc service-name
3. sbe
4. adjacency sip adjacency-name
5. ping-enable
6. ping-interval interval
7. ping-lifetime duration
8. ping-fail-count fail-count
9. exit

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td></td>
</tr>
<tr>
<td>configure terminal</td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td></td>
</tr>
<tr>
<td>sbc service-name</td>
<td>Enters the mode of an SBC service.</td>
</tr>
<tr>
<td>Example:</td>
<td>Use the service-name argument to define the name of the service.</td>
</tr>
<tr>
<td>Router(config)# sbc mysbc</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td></td>
</tr>
<tr>
<td>sbe</td>
<td>Enters the mode of an SBE entity within an SBC service.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc)# sbe</td>
<td></td>
</tr>
</tbody>
</table>
### SIP UAS Failure Detection: Example

In the following configuration example, PING is enabled on each of three adjacencies. A round robin call policy is set so that calls are distributed between the three adjacencies in a weighted random manner. If a UAS is unreachable, calls will be distributed between the remaining two adjacencies.

```
sbc mySBC
  sbe
    adjacency sip CallMgrA
      signaling-address ipv4 88.103.29.100
      remote-address ipv4 200.200.200.0 255.255.255.0
      signaling-peer 200.200.200.118
      ping-enable
        ping-interval 5
        ping-fail-count 3
        ping-lifetime 32
        attach
    adjacency sip CallMgrB
      signaling-address ipv4 88.103.29.100
      remote-address ipv4 200.200.200.0 255.255.255.0
      signaling-peer 200.200.200.117
```

#### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>adjacency sip adjacency-name</td>
<td>Enters the mode of an SBE SIP adjacency. Use the <code>adjacency-name</code> argument to define the name of the service.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe)# adjacency sip sipGW</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>ping-enable</td>
<td>Configures the adjacency to poll its remote peer by sending SIP OPTIONS pings to it and enters the ping option submode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-adj-sip)# ping-enable</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>ping-interval interval</td>
<td>Configures the interval between SIP OPTIONS pings sent to the remote peer.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-adj-sip)# ping-interval 100</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>ping-lifetime duration</td>
<td>Configures the duration for which SBC waits for a response to an options ping for the adjacency.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-adj-sip)# ping-lifetime 100</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>ping-fail-count fail-count</td>
<td>Configures the number of consecutive pings that must fail before the adjacencies peer is deemed to be unavailable.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-adj-sip)# ping-fail-count 10</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>exit</td>
<td>Exits <code>adj-sip</code> mode to <code>sbe</code> mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-adj-sip)# exit</td>
<td></td>
</tr>
</tbody>
</table>
SIP Outbound Flood Protection

SIP Outbound Flood Protection protects other network elements from excessively high valid traffic in unusual situations, such as a protection from a flood of generated BYE messages when a neighboring network element fails.

SIP Outbound Flood Protection sets a maximum rate of outgoing request messages and prevents the rate of outgoing request messages exceeding this maximum rate. If the limit is reached, outgoing requests are failed or dropped instead.

SIP Outbound Flood Protection is an addition to the normal CAC policy mechanisms and does not replace CAC policy. CAC policy allows fine grain control of calls, like, for example, rate limiting of INVITE requests at configurable scopes. SIP Outbound Flood Protection is intended to provide a simple overall rate limit for outgoing requests and is especially useful for requests that currently do not involve CAC policy (such as BYE requests).

Flood protection may be required in the following situations:

- Adjacent network element terminating — If an adjacent network element terminates (either normally or due to error) Cisco Unified Border Element (SP Edition) is likely to detect that the calls that used this element are dead at approximately the same time and attempt to tear the calls down. With many active calls this can generate a flood of BYE requests (normally two BYEs for each call).
Rather than allow these BYE messages to transiently overload other network signaling elements the network administrator may prefer to drop or fail some BYE requests at the Cisco Unified Border Element (SP Edition).

- Local removal of configuration in the Cisco Unified Border Element (SP Edition) — If a SIP adjacency is unconfigured using normal deactivation mode then BYE requests will be sent for all active calls using the adjacency before they are destroyed.

Again it may be desirable for to limit the rate of outgoing requests prevent other network elements getting overloaded.

Use the procedure in this section to configure SIP Outbound Flood Protection:

**SUMMARY STEPS**

1. configure terminal
2. sbc service-name
3. sbe
4. adjacency sip adjacency-name
5. outbound-flood-rate rate
6. exit

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure terminal</td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td>Example: Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 2 sbc service-name</td>
<td>Enters the mode of an SBC service.</td>
</tr>
<tr>
<td>Example: Router(config)# sbc mysbc</td>
<td>Use the service-name argument to define the name of the service.</td>
</tr>
<tr>
<td>Step 3 sbe</td>
<td>Enters the mode of an SBE entity within an SBC service.</td>
</tr>
<tr>
<td>Example: Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td>Step 4 adjacency sip adjacency-name</td>
<td>Enters the mode of an SBE SIP adjacency.</td>
</tr>
<tr>
<td>Example: Router(config-sbc-sbe)# adjacency sip sipGW</td>
<td>Use the adjacency-name argument to define the name of the service.</td>
</tr>
</tbody>
</table>
SIP Over TLS

This section describes the concepts for SIP over Transport Layer Security (TLS). This section contains the following topics:

- Security Configuration on an Adjacency, page 6-21
- SIP Over TLS Overview, page 6-21
- User Agent Server-Side Processing, page 6-24
- Routing Processing, page 6-24
- User Agent Client-Side Processing, page 6-24
- Configurable Mutual TLS Authentication Per Interface, page 6-24
- TLS Transport Parameter in Record-Route Headers, page 6-26
- Configuring SIP Over TLS on Cisco Unified Border Element (SP Edition), page 6-27
- SIP Over TLS Configuration Example, page 6-29
- SIP Over TLS Verification, page 6-30
- Enabling the Conversion of SIPS URIs to SIP URIs on a Trusted-Unencrypted Adjacency, page 6-30

SIP Outbound Flood Protection: Example

The following configuration example sets an outbound flood rate of 100 outbound request signals per second.

```bash
sbc mySBC
  sbc
  adjacency sip CallMgrA
    signaling-address ipv4 88.103.29.100
    remote-address ipv4 200.200.200.0 255.255.255.0
    signaling-peer 200.200.200.118
    outbound-flood rate 100
    attach
```

### Command or Action | Purpose
--- | ---
**outbound-flood-rate rate** | Configures the maximum desired rate of outbound request signals on this adjacency (excluding ACK/PRACK requests) in signals per second.

**Example:**
```
Router(config-sbc-sbe-adj-sip)# outbound-flood-rate 1000
```

**exit** | Exits `adj-sip` mode to `sbe` mode.

**Example:**
```
Router(config-sbc-sbe-adj-sip)# exit
```
Security Configuration on an Adjacency

You can independently configure client and server security support on a SIP adjacency, using the following options:

- **Untrusted**—Specifies that this adjacency is not secured by any means. Only unsecured calls (not the calls to SIPS URIs) are made out of this adjacency.
- **Untrusted-Encrypted**—Specifies that the adjacency is untrusted and SSL/TLS encryption is used.
- **Trusted-Encrypted**—Specifies that the encrypted signaling is used to ensure security on this adjacency. The default certificate and key of the router are used for encryption. Only secure calls (calls to SIPS URIs) are made out of this adjacency.
- **Trusted-Unencrypted**— Specifies that a non-encryption mechanism is used to guarantee secure signaling for all messages on this adjacency. For example, this mechanism could be a single trusted physical link. Either secure or unsecured calls are made out of this adjacency. This configuration allows endpoints that do not support encryption to participate in secure SIP calls.

SIP Over TLS Overview

SIP Over Transport Layer Security (TLS) encryption provides a secure, encrypted transport to carry all SIP messages from the caller to the callee's domain. From there, the request is sent securely to the callee. Cisco Unified Border Element (SP Edition) provides the following support for SIP Over TLS:

- Secured SIP calls can flow through Cisco Unified Border Element (SP Edition).
- A SIP adjacency can be secured by encryption or by another mechanism (for example, a single trusted physical-layer link or an interface to a trusted network).
- Inbound and outbound connections are immediately closed if a remote peer attempts to use encryption when encryption is not supported.
- Inbound and outbound connections are immediately closed if a remote peer fails to use encryption when encryption is required.
- You can view the level of security support configured for a given SIP adjacency by using the `show sbc sbc-name sbe adjacencies adj name detail` command.
- Calls received on untrusted adjacencies are not routed over outbound secure-encrypted adjacencies.
- Adjacencies secured by means of encryption can listen by default on port 5061. The port is configured to a different value.
- The fully-qualified domain name (FQDN) in the certificate offered by the remote peer is checked against the domain from which the request is received. The signal is dropped if the two do not match.
- Advanced Encryption Standard (AES) 128-bit Secure Hash Algorithm (SHA) is supported.
- The PKI HA updates the standby router with the certificate and trustpoint configuration changes.

The following are main security factors that are used in routing or rejecting a call:

- Calls to a SIPS URI must be secure. Calls to a SIP URI do not have to be secure.
- Signals received on a trusted adjacency are considered secure. Signals received on an untrusted adjacency are considered unsecured.
The following security factors apply to untrusted encrypted adjacencies:

- Secure calls may not be received on untrusted adjacencies of any type.
  - Cisco Unified Border Element (SP Edition) allows unsecured calls to be received over the untrusted encrypted adjacency.
  - Cisco Unified Border Element (SP Edition) rejects secured call that it receives over the untrusted encrypted adjacency.

- Secure calls cannot be routed to untrusted adjacencies.
  - Cisco Unified Border Element (SP Edition) can route unsecured calls over the untrusted encrypted adjacency.
  - Cisco Unified Border Element (SP Edition) does not route secured calls over the untrusted encrypted adjacency.

Table 6-1 and Table 6-2 summarize how Cisco Unified Border Element (SP Edition) handles inbound and outbound calls based on the call type, trust relationship, and encryption.

**Table 6-1 Inbound Call Policy**

<table>
<thead>
<tr>
<th>SIP Call Type</th>
<th>Trusted Adjacency</th>
<th></th>
<th>Untrusted Adjacency</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Encrypted</td>
<td>Unencrypted</td>
<td>Encrypted</td>
<td>Unencrypted</td>
</tr>
<tr>
<td>Secure SIP</td>
<td>Allow</td>
<td>Allow</td>
<td>Reject</td>
<td>Reject</td>
</tr>
<tr>
<td>Unsecured SIP</td>
<td>Allow</td>
<td>Allow</td>
<td>Allow</td>
<td>Allow</td>
</tr>
</tbody>
</table>

**Table 6-2 Outbound Call Policy**

<table>
<thead>
<tr>
<th>SIP Call Type</th>
<th>Trusted Adjacency</th>
<th></th>
<th>Untrusted Adjacency</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Encrypted</td>
<td>Unencrypted</td>
<td>Encrypted</td>
<td>Unencrypted</td>
</tr>
<tr>
<td>Secure SIP</td>
<td>Allow</td>
<td>Allow</td>
<td>Reject</td>
<td>Reject</td>
</tr>
<tr>
<td>Unsecured SIP</td>
<td>Allow</td>
<td>Allow</td>
<td>Allow</td>
<td>Allow</td>
</tr>
</tbody>
</table>

Table 6-3 and Table 6-4 summarize how Cisco Unified Border Element (SP Edition) handles inbound and outbound registrations based on the registration type, trust relationship, and encryption.

**Table 6-3 Inbound Registration Policy**

<table>
<thead>
<tr>
<th>SIP Registration Type</th>
<th>Trusted Adjacency</th>
<th></th>
<th>Untrusted Adjacency</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Encrypted</td>
<td>Unencrypted</td>
<td>Encrypted</td>
<td>Unencrypted</td>
</tr>
<tr>
<td>Secure SIP</td>
<td>Allow</td>
<td>Allow</td>
<td>Reject</td>
<td>Reject</td>
</tr>
<tr>
<td>Unsecured SIP</td>
<td>Allow</td>
<td>Allow</td>
<td>Allow</td>
<td>Allow</td>
</tr>
</tbody>
</table>
For the SBC to be able to forward Secure SIP (SIPS) registrations to a trusted-unencrypted adjacency, all the following conditions must be met:

- The source adjacency must have a non-IP Multimedia Subsystem (non-IMS) or non-IMS access adjacency profile that specifies tracking of the registration state.
- The destination adjacency must have a non-IMS adjacency profile.
- The destination adjacency must be configured to accept a SIPS URI registration. This procedure is explained in the "Enabling the Conversion of SIPS URIs to SIP URIs on a Trusted-Unencrypted Adjacency" section on page 6-30.

When the SBC forwards secure registrations to a trusted-unencrypted adjacency that meets these conditions, the outbound registration is modified as follows:

- The Address of Record (AoR) in the To and From headers is converted from a SIPS URI to a SIP URI.
- The Request URI is converted from a SIPS URI to a SIP URI. Note that the Request URI may not be identical to the AoR.
- The URIs in the Contact headers are converted from SIPS to SIP.
- The URIs in Record Route headers are passed through unchanged. Note that according to RFC 3261, Record Route headers must be ignored on receipt if they are present in REGISTER messages.
- The URIs in other SIP headers are passed through unchanged.

The SBC rejects registrations that contain a mix of SIP URIs and SIPS URIs in their AoRs and contacts. On receipt of the REGISTER response, the SBC reverses the changes and passes back SIPS URIs in the response forwarded to the endpoint.

The following are restrictions for this feature:

- There is no change in the processing of non-INVITE messages, such as SUBSCRIBE, NOTIFY, and PUBLISH, by the SBC. For these messages, the SBC does not convert SIPS URIs to SIP URIs.
- The SBC does not support registrations to trusted-unencrypted adjacencies in scenarios where either the inbound adjacency or the outbound adjacency has an IMS profile.

---

### Table 6-4 Outbound Registration Policy

<table>
<thead>
<tr>
<th>SIP Registration Type</th>
<th>Trusted Adjacency</th>
<th>Untrusted Adjacency</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Encrypted</td>
<td>Unencrypted</td>
</tr>
<tr>
<td>Secure SIP</td>
<td>Allow</td>
<td>Allow or Reject (depending on the configuration)</td>
</tr>
<tr>
<td>Unsecured SIP</td>
<td>Allow</td>
<td>Allow</td>
</tr>
</tbody>
</table>

---

**Note**

The SBC rejects registrations that contain a mix of SIP URIs and SIPS URIs in their AoRs and contacts. On receipt of the REGISTER response, the SBC reverses the changes and passes back SIPS URIs in the response forwarded to the endpoint.
User Agent Server-Side Processing

Inbound requests are marked according to two factors: whether the caller is trusted, and whether the call is intended for a secure target.

The caller-trust is determined in the following ways:

- SIP requests arriving over trusted adjacencies are marked as trusted.
- SIP requests arriving over untrusted adjacencies are marked as untrusted.

Desired target security is determined in the following ways:

- Requests for SIPS URIs are marked to require the outbound-security.
- Requests for SIP URIs are marked not to require the outbound-security.

Inbound requests are rejected if the caller is untrusted and the target requires security. All other combinations are forwarded to routing processing.

Routing Processing

The Routing Policy System (RPS) policy determines where the requests are routed next, with the following default behaviors:

- If a call requires the outbound security, the RPS considers only the trusted outbound adjacencies.
- If a call does not require the outbound security, the RPS considers only untrusted or trusted-unencrypted outbound adjacencies.

If the RPS is unable to find a suitable outbound adjacency for a call, the call is rejected.

User Agent Client-Side Processing

Outbound adjacencies preserve the URI scheme of the original request, ensuring that if a call is originally targeted at a SIPS URI, it is sent out to a SIPS URI. Or, if the call is originally targeted at a SIP URI, it is sent out to a SIP URI.

Upon receipt of 3xx class responses and target-refresh indications, the contact set is examined. Untrusted adjacencies do not permit the target of the call to be rerouted to a SIPS target. Likewise, trusted adjacencies do not permit the target of the call to be rerouted to a SIP target. If this is attempted by the remote peer, the call is brought down.

Configurable Mutual TLS Authentication Per Interface

The Configurable Mutual TLS Authentication Per Interface feature helps you to configure unilateral or mutual TLS authentication on a per adjacency basis for SIP over TLS calls.

In a SIP over TLS call, SBC can either be a TLS client side or TLS server side. This feature is relevant only when the SBC is a TLS server side.

While negotiating a TLS connection, the server side sends its certificate to the client side to perform the server authentication. After the server authentication, the server may require a certificate from the client for client authentication. When client authentication is not used, the authentication is referred to as a unilateral authentication. When both — server and client — requires authentication, then it is referred to as a mutual authentication.
The message flow diagram, Figure 6-5, illustrates the negotiation process of a TLS connection. The bold line represents the messages required when mutual authentication is enabled on the server side:

**Figure 6-5 Mutual TLS Authentication Message Flow Diagram**

When SBC acts as TLS client side, it can automatically negotiate with the server side to perform the client authentication. But when SBC acts as TLS server side, you need to configure SBC so that SBC can decide whether to send a CertificateRequest message to the client side to get the client's certificate to do client authentication.

Use the `tls mutual-authentication` command to configure mutual authentication.

### Restrictions and Limitations—Configurable Mutual TLS Authentication

- The configuration on a SIP adjacency cannot be modified while the adjacency is attached.
- The security configuration of the adjacency must be trusted encrypted or untrusted encrypted.
- Multiple TLS-enabled adjacencies that use the same local address and port must have the same configuration. Otherwise, the configuration will be rejected and an error message will be displayed on the console.
- Configuring trust points on a per adjacency basis is not possible because SBC uses global trust points to validate peer's certification. This limitation will not pose any limitation for certificate verification on SBC because, SBC automatically searches for a matching certificate from its global trust points.
SBC only supports one certificate while SBC is a TLS server side. It is not possible to configure different certificates for each adjacency. The certificate is picked from the primary trust point.

Certificate chain is not synchronized in SSO config-sync redundancy mode and hence the TLS certificates are not replicated to the standby. The incoming TLS calls might fail because of non-availability of TLS certificates.

**TLS Transport Parameter in Record-Route Headers**

This feature allows you to add a `transport=tls` parameter to the SBC-originated Contact and Record-Route headers when using TLS. This feature is applicable when the security for the SBC inbound adjacency is configured as untrusted-encrypted.

The transport=tls parameter was deprecated in RFC3261 for better interoperability. With the implementation of RFC3261, the Contact and Record-Route header of 200(INVITE), back to caller, would use SIP URI without the transport parameter such as `Contact: <sip:192.168.1.1:5060>`, `Record-Route: <sip:192.168.1.1:5060;lr>`. Because of this, the subsequent mid-dialog requests—re-INVITE—are sent using TCP or UDP based on the SIP URI instead of TLS. Since SBC is expecting a TLS message on the port, the call is dropped.

Figure 6-6 shows the message flow where the SIP call is received over TLS, but the call was dropped. The ACK in response to the 200OK (TLS) message is sent from the SIP to SBC using TCP or UDP.

![Figure 6-6 Message Flow During a SIP Call Over TLS](image)

To avoid call drops, the caller is forced to use the TLS transport for the ACK by adding the transport=tls parameter. This feature is controlled on a per adjacency basis.

Use `header-name [contact [add [tls-param]] | from{passthrough} | to{passthrough}] command in (config-sbc-sbe-adj-sip) mode to configure the transport=tls parameter in the Contact and Record-Route header.
## Configuring SIP Over TLS on Cisco Unified Border Element (SP Edition)

Use the procedure in this section to configure SIP over TLS on Cisco Unified Border Element (SP Edition):

### SUMMARY STEPS

1. `configure terminal`
2. `sbc service-name`
3. `sbe`
4. `adjacency sip adjacency-name`
5. `security trusted-encrypted`
6. `redirect-mode {pass-through | recurse}`
7. `authentication nonce`
8. `signaling-address ipv4 ipv4_IP_address`
9. `signaling-port port-num`
10. `remote-address ipv4 ip-address ip-mask`
11. `signaling-peer peer-name`
12. `signaling-peer-port port-num`
13. `dbe-location-id dbe-location-id`
14. `reg-min-expiry period`
15. `attach force [abort | normal]`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>configure terminal</code></td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>sbc service-name</code></td>
<td>Enters the mode of an SBC service. Use the <code>service-name</code> argument to define the name of the service.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# sbc mysbc</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>sbe</code></td>
<td>Enters the mode of an SBE entity within an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> <code>adjacency sip adjacency-name</code></td>
<td>Enters the mode of an SBE SIP adjacency. Use the <code>adjacency-name</code> argument to define the name of the service.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe)# adjacency sip sipGW</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>security trusted-encrypted</td>
<td>Configures transport-level security on a Session Initiation Protocol (SIP) adjacency.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-adj-sip)# security trusted-encrypted</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>redirect-mode (pass-through</td>
<td>Configures the behavior of SBC on receipt of a 3xx response to an invite from the SIP adjacency.</td>
</tr>
<tr>
<td></td>
<td>recurse)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-adj-sip)# redirect-mode recurse</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>authentication nonce timeout</td>
<td>Configures the authentication nonce timeout for a SIP adjacency.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbe-adj-sip)# authentication nonce timeout 10</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>signaling-address ipv4 ipv4_IP_address</td>
<td>Defines the local IPv4 signaling address of a SIP or an H.323 adjacency.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-adj-sip)# signaling-address ipv4 10.10.10.10</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>signaling-port port-num</td>
<td>Defines the local port of signaling address of a SIP adjacency.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-adj-sip)# signaling-port 5000</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>remote-address ipv4 ip-address ip-mask</td>
<td>Configures a SIP adjacency to restrict the set of remote signaling peers that can be contacted over the adjacency to those with the given IP address prefix.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-adj-sip)# remote-address ipv4 36.36.36.20 255.255.255.0</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>signaling-peer peer-name</td>
<td>Configures a SIP adjacency to use the given remote signaling-peer.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-adj-sip)# signaling-peer 10.1.2.3</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>signaling-peer-port port-num</td>
<td>Configures a SIP adjacency to use the given remote signaling-peer’s port.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-adj-sip)# signaling-peer-port 123</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>dbe-location-id dbe-location-id</td>
<td>Configures an adjacency to use a given media gateway DBE location when routing media.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-adj-sip)# dbe-location-id 1</td>
<td></td>
</tr>
</tbody>
</table>
Chapter 6  Implementing Adjacencies on Cisco Unified Border Element (SP Edition)

SIP Over TLS

The following example shows a SIP over TLS configuration:

```
crypto pki trustpoint CA
  enrollment terminal
  serial-number
  subject-name ST=Some-State, C=AU, O=Internet Widgits Pty Ltd
  revocation-check none
rsakeypair the_default

crypto pki certificate chain CA
  certificate 01
    308201D7 30820140 A0030201 02020101 300D0609 2A864886 F70D0101 04050030
    46310B30 09060350 0040B013 53424330 310B3009 06035504 07130253 4A310E30
    0C060355 040A1305 43495343 4F310C30 0A060355
    040B1303 53424330 31070D30 39032300 35313030 3832385A 170D3110 30323035
    31303038 32385A30 45310E30 06035504 0801303A 536F6D65 2D537461 74656110
    30090603 55040613 02415531 21301F06 0355040A 1318496E 7465726E 65742057
    69646769 74732050 7479204C 7464305C 300D0609 2A864886 F70D0101 01050003
    4B003008 241000DC 18647810 40762B06 3F5D0106 0E3C0001 0C130103 300D0609
    2A864886 F70D0101 01050003
quit

certificate ca 00F2D75C678DC7F7F2
  3082021A 00302010 02020101 300D0609 2A864886 F70D0101 04050030
    46310B30 09060350 0040B013 53424330 31070D30 39032300 35313030 3832385A
    170D3110 30323035 31303038 32385A30 45310E30 06035504 0801303A 536F6D65
    2D537461 74656110 30090603 55040613 02415531 21301F06 0355040A 1318496E
    7465726E 65742057 69646769 74732050 7479204C 7464305C 300D0609 2A864886
    F70D0101 01050003
quit
```
The Cisco Unified Border Element (SP Edition) configuration example is illustrated here.

Router# configure
Router(config)# sbc sbc-3
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency sip adj1
Router(config-sbc-sbe-adj-sip)# security trusted-encrypted
Router(config-sbc-sbe-adj-sip)# redirect-mode pass-through
Router(config-sbc-sbe-adj-sip)# authentication nonce timeout 300
Router(config-sbc-sbe-adj-sip)# signaling-address ipv4 10.130.10.25
Router(config-sbc-sbe-adj-sip)# signaling-port 5060
Router(config-sbc-sbe-adj-sip)# remote-address ipv4 10.74.49.145 255.255.255.255
Router(config-sbc-sbe-adj-sip)# signaling-peer 10.74.49.145
Router(config-sbc-sbe-adj-sip)# signaling-peer-port 5060
Router(config-sbc-sbe-adj-sip)# dbe-location-id 4294967295
Router(config-sbc-sbe-adj-sip)# reg-min-expiry 3000
Router(config-sbc-sbe-adj-h323)# attach

SIP Over TLS Verification

Use the following commands to check certificates on the node:

- `show crypto pki certificates`—displays information about your certificate, the certification authority certificate (CA), and any registration authority (RA) certificates.
- `show crypto key pubkey-chain rsa`—enters public key configuration mode (so you can manually specify and show other devices' RSA public keys).
- `show crypto key mypubkey rsa`—displays the RSA public keys of your router.

Enabling the Conversion of SIPS URIs to SIP URIs on a Trusted-Unencrypted Adjacency

Use the procedure described in this section to enable the conversion of SIPS URIs to SIP URIs on a trusted-unencrypted adjacency. Performing this procedure is one of the requirements for configuring the SBC to forward secure registrations to a trusted-unencrypted adjacency. See Chapter 6 > SIP Over TLS Overview? section on page 6-21 for more information about this feature.

SUMMARY STEPS

1. configure terminal
2. sbc sbc-name
3. sbe
4. adjacency sip adjacency-name
5. security trusted-unencrypted
6. registration unencrypted-convert
7. end
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>configure terminal</code></td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>Router# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>sbc</code> <code>sbc-name</code></td>
<td>Enters the SBC service mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>Router(config)# sbc mysbc</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>sbe</code></td>
<td>Enters the SBE configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>Router(config-sbc)# sbe</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> <code>adjacency sip</code> <code>adjacency-name</code></td>
<td>Enters the mode of an SBE SIP adjacency.</td>
</tr>
<tr>
<td>Step 5</td>
<td></td>
</tr>
<tr>
<td><code>Router(config-sbc-sbe)# adjacency sip</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> <code>security trusted-unencrypted</code></td>
<td>Configures transport-level security on the adjacency.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>Router(config-sbc-sbe-adj-sip)# security</code></td>
<td>trusted-unencrypted</td>
</tr>
<tr>
<td><strong>Step 6</strong> <code>registration unencrypted-convert</code></td>
<td>Enables the conversion of SIPS URIs to SIP URIs on the trusted-unencrypted adjacency.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>Router(config-sbc-sbe-adj-sip)# registration</code></td>
<td>unencrypted-convert</td>
</tr>
<tr>
<td><strong>Step 7</strong> <code>end</code></td>
<td>Returns to the privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>Router(config-sbc-sbe-adj-sip)# end</code></td>
<td></td>
</tr>
</tbody>
</table>

In the following example, the output of the `show sbc adjacencies` command shows that conversion of SIPS URIs to SIP URIs is enabled:

```
Router# show sbc MySBC sbe adjacencies ADJ1 detail
SBC Service MySBC
   Adjacency ADJ1 (SIP)
     Status:    Attached
     Signaling address: 192.0.2.36.1:5060, VRF sidd_sipp1
     IPsec server port: 0
     Signaling-peer: 192.0.2.37.1:5060 (Default)

Media Bypass Tag List:
   Tag 1:   tag1
```
SIP Peer Availability Detection

The SBC supports the SIP peer availability detection (OPTIONs ping) functionality. The SBC periodically sends an OPTION request to a configured peer. If the peer fails to respond to a set number of OPTION requests, the peer is declared dead, and the calls are routed through other peers.

To avoid congestion, when ping suppression is enabled, and if signaling traffic exchange between peers is active, the OPTIONS pings are not used to check peer availability.

Restrictions for SIP Peer Availability Detection

The SIP Peer Availability Detection feature has the following restrictions:

- The OPTION requests will use the SIP method congestion response codes.
- If the number of OPTIONS messages to the peer are reduced, the time taken to detect dead peer by the SBC increases substantially.

Configuring SIP Peer Availability Detection

Use the procedure described in this section to configure the detection of SIP peer availability:

SUMMARY STEPS

1. configure terminal
2. sbc service-name
3. sbe
4. adjacency sip adjacency-name
5. ping-enable
6. ping-bad-rsp-codes ranges
7. ping-suppression
8. exit
9. end
10. show sbc sbc-name sbe adjacencies adjacency-name Detail
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc service-name</td>
<td>Enters the mode of an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# sbc mysbc</td>
<td>Use the <code>service-name</code> argument to define the name of the service.</td>
</tr>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Enters the mode of an SBE entity within an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> adjacency sip adjacency-name</td>
<td>Enters the mode of an SBE SIP adjacency.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe)# adjacency sip sipGW</td>
<td>Use the <code>adjacency-name</code> argument to define the name of the service.</td>
</tr>
<tr>
<td><strong>Step 5</strong> ping-enable</td>
<td>Configures the adjacency to poll the adjacency's remote peer by sending SIP OPTION pings to it, and enters the Ping option submode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-adj-sip)# ping-enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> ping-bad-rsp-codes ranges</td>
<td>Configures the congestion response codes on a SIP adjacency by sending SIP OPTION pings to the adjacency.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-adj-sip-ping)# ping-bad-rsp-codes ranges 300,398</td>
<td>Use the <code>ranges</code> argument to specify the response code range (The range can be 300 to 399).</td>
</tr>
<tr>
<td><strong>Step 7</strong> ping-suppression options</td>
<td>(Optional) Configures the SBC to ping, when required, on a SIP adjacency.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-adj-sip-ping)# ping-suppression odd-request</td>
<td><code>options</code> specifies one of the following strings used for ping suppression:</td>
</tr>
<tr>
<td></td>
<td>- <code>ood-request</code>—The SBC considers a peer as reachable when an out-of-dialog or dialog-creating request is received, excluding the OPTIONS and REGISTER messages.</td>
</tr>
<tr>
<td></td>
<td>- <code>ood-response</code>—The SBC considers a peer as reachable when an out-of-dialog or dialog-creating 2xx response is received, excluding OPTIONS and REGISTER messages.</td>
</tr>
<tr>
<td></td>
<td>- <code>ind-request</code>—The SBC considers a peer as reachable when an in-dialog request is received.</td>
</tr>
<tr>
<td></td>
<td>- <code>ind-response</code>—The SBC considers a peer as reachable when an in-dialog 2xx response is received.</td>
</tr>
</tbody>
</table>
Chapter 6  Implementing Adjacencies on Cisco Unified Border Element (SP Edition)

Redundant Peer Addresses

The SBC may be required to interoperate with peer SIP devices, such as the PGW 2200 softswitch, which can start signaling from a different IP address following a VLAN failure when operating with redundant hosts on separate VLANs (for example, geographically separated hosts). Peer SIP devices, such as the PGW 2200 softswitch, have the following high availability (HA) strategies that include both VLAN and host redundancy:

- In a standard failover scenario, when one host fails, a backup takes over. This backup also takes over the virtual IP address used for SIP communication. The call state is maintained, and the failover is made invisible to the SBC.
- In a scenario where a VLAN failure occurs, the PGW 2200 softswitch host, which inter interoperates with the SBC, starts using an interface in a different VLAN. Because you cannot transfer a virtual IP address between VLANs, the IP address for SIP communication changes.
- In some networks, the primary and backup hosts are geographically redundant and unable to share a VLAN. Therefore, you cannot transfer a virtual IP address between the hosts, so the IP address being used for SIP communication changes if the primary host fails.

---

### Example

The following example shows how to configure the congestion response codes on a SIP adjacency by sending SIP OPTIONS pings:

```
Example:
Router# configure terminal
Router(config)# sbc mySbc
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency sip SipAdj1
Router(config-sbc-sbe-adj-sip)# ping-enable
Router(config-sbc-sbe-adj-sip-ping)# ping-bad-rsp-codes ranges 300,398
Router(config-sbc-sbe-adj-sip-ping)# exit
```

---

### Command or Action | Purpose
--- | ---
**Step 8** exit | Exits the adj-sip-ping mode, and moves to the adj-sip mode.
Example: `Router(config-sbc-sbe-adj-sip-ping)# exit` |  |
**Step 9** end | Exits the SBE mode and returns to the Privileged EXEC mode.
Example: `Router(config-sbc-sbe)# end` |  |
**Step 10** show sbc sbc-name sbe adjacencies adjacency-name detail | Displays the details pertaining to the specified adjacency.
Example: `Router# show sbc mySbc sbe adjacencies sipGW detail` |  |
Chapter 6  Implementing Adjacencies on Cisco Unified Border Element (SP Edition)

Figure 6-7 illustrates a redundant topology.

Redundant Peer Addresses

When peer IP addresses change, the call state on the corresponding peer device continues to be maintained, and key SIP parameters, such as the dialog tags, the Contact header, and route set, are unchanged. However, the Contact header is incorrect, and does not contain the new peer IP address.

The Redundant Peer Addresses feature allows the IP address of its peer device to change, and supports the following functionalities on SBC:

- Accepts incoming SIP messages from any of the redundant IP addresses.
- Ignores discrepancies between the IP addresses specified in the SIP Contact header (or other SIP headers) and the actual IP address being used by the peer.
- Pings each peer address to monitor the active addresses and sends the outgoing SIP messages destined for the peer to an IP address that is currently active.
- Configures the SBC with multiple redundant IP addresses for a SIP peer device that is not contained within a single remote address mask.
- Supports the following modes of operation that is configurable for each adjacency:
  - The SBC switches peer IP address when a higher priority address becomes active, even if the current address does not fail.
  - Elects a current destination for each adjacency, choosing the peer IP address with the highest-priority that is currently active, and continues to use that destination until it is detected to have failed, at which point the election process is repeated.
- Uses a single local IP address, port, and VPN for all communication with every peer IP address.
Restrictions for Redundant Peer Addresses

The Redundant Peer Addresses feature has the following restrictions:

- This is a signaling-only feature.
- Alternative redundant addresses for a peer cannot be automatically detected, and must therefore, be configured using the `ping-enable` command.
- The main peer address in an adjacency share the same priority values, ranging from 1 to 6, as the redundant peer addresses.
- A single load-balancing method is provided. The SBC selects the active peer IP address with the highest configured priority for all the outgoing SIP requests.
- The source address of fast-REGISTER requests cannot be changed.
- If a SIP request is sent to a peer address, and no response is received, the SBC subsequently detects that the peer address has failed. However, the destination address of the SIP request does not change, and it is retried to the failed address. New requests are sent to an active address.
- The destination addresses and ports configured for a given adjacency are not available in message editing configuration. Therefore, there is no per-destination equivalent for the existing signaling-peer and signaling-peer-port header filtering syntax.
- The optimization to send only pings when required (ping suppression) cannot be configured on the adjacencies facing redundant peers.

Configuring Redundant Peer Addresses

Use the procedure described in this section to configure redundant peer addresses:

**SUMMARY STEPS**

1. `configure terminal`
2. `sbc service-name`
3. `sbe`
4. `adjacency sip adjacency-name`
5. `no attach`
6. `force-signaling-peer all`
7. `ping-enable`
8. `exit`
9. `redundant peer index`
10. `address address`
11. `port port`
12. `network {IPv4 address netmask | IPv6 address netmask}`
13. `priority priority`
14. `activate`
15. `exit`
16. `signaling-peer-switch {always | fail}`
17. `signaling-peer-priority priority`
18. `exit`
19. `end`
20. `show sbc sbc-name sbe adjacencies adjacency-name peers`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>configure terminal</code></td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>sbc service-name</code></td>
<td>Enters the mode of an SBC service. Use the <code>service-name</code> argument to define the name of the service.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# sbc mysbc</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>sbe</code></td>
<td>Enters the mode of an SBE entity within an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> <code>adjacency sip adjacency-name</code></td>
<td>Enters the mode of an SBE SIP adjacency. Use the <code>adjacency-name</code> argument to define the name of the service.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe)# adjacency sip sipGW</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> <code>no attach</code></td>
<td>Detaches the adjacency from an account on the SBE. <strong>Note</strong> The adjacency must be detached before adding or removing a redundant peer.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-adj-sip)# no attach</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> <code>force-signaling-peer all</code></td>
<td>Forces SIP messages for both in-call requests and out-of-call requests to go to the configured signaling peer.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-adj-sip)# force-signaling-peer all</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> <code>ping-enable</code></td>
<td>Configures the adjacency to poll its remote peer by sending SIP OPTIONS pings to it, and enters the ping option submode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-adj-sip)# ping-enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> <code>exit</code></td>
<td>Exits the <code>adj-sip-ping</code> mode, and moves to <code>adj-sip</code> mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-adj-sip-ping)# exit</td>
<td></td>
</tr>
</tbody>
</table>
## Redundant Peer Addresses

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>redundant peer index</td>
<td>Enters the mode of an SBE SIP adjacency peer to configure an alternative signaling peer for the adjacency. You can specify the index number of the peer, ranging from 1 to 5.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>Router(config-sbc-sbe-adj-sip)# redundant peer 1</code></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>address address</td>
<td>Configures either an IP address or a host name to act as the redundant peer.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>Router(config-sbc-sbe-adj-sip)# no address</code></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>port port</td>
<td>Configures a port for the redundant peer.</td>
</tr>
<tr>
<td>Note</td>
<td>By default, 5060 port is used.</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td><code>Router(config-sbe-adj-sip-peer)# port 2</code></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>network {IPv4 address netmask</td>
<td>IPv6 address netmask}</td>
</tr>
<tr>
<td>Example:</td>
<td><code>Router(config-sbc-sbe-adj-sip-peer)# network ipv4 33.33.36.2 255.255.255.0</code></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>priority priority</td>
<td>Configures the redundant peer's priority. The range is from 1 to 6.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>Router(config-sbc-sbe-adj-sip-peer)# priority 1</code></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>activate</td>
<td>Activates the redundant signaling peer.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>Router(config-sbc-sbe-adj-sip-peer)# activate</code></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>exit</td>
<td>Exits the <code>adj-sip-peer</code> mode, and moves to <code>adj-sip</code> mode.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>Router(config-sbc-sbe-adj-sip-peer)# exit</code></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>signaling-peer-switch {always</td>
<td>fail}</td>
</tr>
<tr>
<td>Example:</td>
<td><code>Router(config-sbc-sbe-adj-sip)# signaling-peer-switch always</code></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>signaling-peer-priority priority</td>
<td>Configures the priority of a signaling peer on a SIP adjacency. The range is from 1 to 6.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>Router(config-sbc-sbe-adj-sip)# signaling-peer-priority 1</code></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>exit</td>
<td>Exits the <code>adj-sip</code> mode, and moves to <code>sbe</code> mode.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>Router(config-sbc-sbe-adj-sip)# exit</code></td>
<td></td>
</tr>
</tbody>
</table>
## Redundant Peer Addresses

The following example shows a redundant peer addresses configuration:

```bash
sbc mat
sbe
  adjacency sip SIPPA
  force-signaling-peer all
  signaling-peer-switch on-fail
  inherit profile preset-access
  signaling-address ipv4 1.0.0.10
  statistics method summary
  signaling-port 5068
  remote-address ipv4 1.0.0.0 255.0.0.0
  signaling-peer 1.0.0.3
  signaling-peer-priority 6
  signaling-peer-port 5068
  registration rewrite-register
  registration target address 1.0.0.3
  registration target port 5068
  redundant peer 1
    network ipv4 5.5.5.5 255.255.255.255
    address 5.5.5.5
    priority 2
    activate
  redundant peer 2
    network ipv4 22.22.22.22 255.255.255.255
    address 22.22.22.22
    port 2222
    priority 3
    ping-enable
    attach
```

### Redundant Peer Addresses Verification

Use the following commands to verify the peers:

- **show sbc sbe adjacencies detail**—Displays detailed configuration of a SIP adjacency.
- **show sbc sbe adjacencies peer**—Lists the configured peers for an adjacency.
- **show sbc sbe all-peers**—Displays a peer’s information.
Implementing Cisco Unified Border Element (SP Edition) Policies

A Cisco Unified Border Element (SP Edition) policy is a set of rules that define how the Cisco Unified Border Element (SP Edition) treats different kinds of voice over IP (VoIP) events. A Cisco Unified Border Element (SP Edition) policy allows you to control the VoIP signaling and media that passes through the Cisco Unified Border Element (SP Edition) at an application level.

Note
From Cisco IOS XE Release 2.4, configuration of policies is supported in the unified model. Enhancements to this feature have been introduced in later releases.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).

For a complete description of the commands used in this chapter, refer to Cisco Unified Border Element (SP Edition) Command Reference: Unified Model at:

For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

Feature History for Implementing Cisco Unified Border Element (SP Edition) Policies

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 2.4</td>
<td>This feature was introduced on the Cisco IOS XR along with support for the unified model.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 2.5</td>
<td>Subscriber Policy support, Regular expression based routing support, SIP trunk-group ID routing support, and the SIP media line removal feature were added on the Cisco ASR 1000 Series Routers. Support for H.323 call routing features: H.323 Hunting and multiARQ hunting, Picking a next Hop in Routing Policy, Support for H.323 Addressing, DNS Name Resolution, Number Validation and Editing, Load Balancing, and Inter-VPN Calling were added on the Cisco ASR 1000 Series Routers.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 2.6</td>
<td>(Source) Number Analysis feature updated to include source number table and source prefix table.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 3.1S</td>
<td>Support for Asymmetric payload types was added.</td>
</tr>
</tbody>
</table>
Contents

Chapter 7  Implementing Cisco Unified Border Element (SP Edition) Policies

This chapter contains the following sections:

- Prerequisites for Implementing Policies, page 7-2
- Restrictions, page 7-3
- Information About Implementing Policies, page 7-3
- Message, Policy, and Subscriber Statistics, page 7-50
- Administrative Domains, page 7-59
- Asymmetric Payload Types, page 7-60
- How to Implement Policies, page 7-65
- Configuring Asymmetric Payload Types, page 7-143
- Limiting Resource Usage, page 7-145
- Configuration Examples for Implementing Policies, page 7-156

Prerequisites for Implementing Policies

The following prerequisites are required to implement Cisco Unified Border Element (SP Edition) policies:

Cisco IOS XE Release 3.2S  The Number Analysis feature was updated to include source address manipulation. The following number analysis tables were changed:
- na-src-number-table was changed to na-src-address-table.
- na-dst-number-table was changed to na-dst-address-table.
- na-dst-number-attr-table was changed to na-carrier-id-table.
- first-number-analysis-table was renamed as first-inbound-na-table.

Also, first-outbound-na-table was introduced, active-call-policy-set was renamed as call-policy-set default, and active-cac-policy-set was renamed as cac-policy-set global.

Administrative domains were introduced.

The copy and swap procedure for Call Admission Control (CAC) and call policy sets was introduced.

The Multiple CAC Averaging Period feature was added.

The Privacy Service feature was added.

The Multiple SBC Media Bypass feature was added.

Cisco IOS XE Release 3.3S  Message, Policy, and Subscriber Statistics enhancements were added.

Cisco IOS XE Release 3.4S  The Limiting Resource Usage feature was added.

Cisco IOS XE Release 3.5S  CAC-related enhancements were introduced. The branch command has been introduced as an alternative to the caller and callee command pair in some configuration scenarios.

Cisco IOS XE Release 3.6S  The Common IP Address Media Bypass feature was added.
Before implementing policies, Cisco Unified Border Element (SP Edition) must already be configured.

Restrictions

The following restrictions apply when you implement routing policies on the Cisco Unified Border Element (SP Edition):

- H.323 protocols are not supported in Cisco IOS XE Release 2.4 and earlier.
- Regular expression matching is only supported for text user names and domain names in source or destination URIs for SIP calls. Regular expression matching for telephone numbers used in H.323 calls is not supported.
- SBC does not allow addition, modification, or removal of trunk-group ID (TGID) information before call routing occurs.
- SBC does not allow regular expression matching when performing TGID routing.

Information About Implementing Policies

A policy is a set of rules that define how the Cisco Unified Border Element (SP Edition) treats different kinds of VoIP events. A Cisco Unified Border Element (SP Edition) policy allows you to control the VoIP signaling and media that passes through Cisco Unified Border Element (SP Edition) at an application level. Figure 7-1 shows an overview of policy control flow.

Figure 7-1  Policy Control Overview

Number analysis and routing are configured in one type of configuration set, admission control is configured in another.

Number analysis (NA) determines whether a set of source digits or dialed digits represents a valid telephone number (based on number validation, number categorization, or digit manipulation). Call routing determines the VoIP signaling entity to which a signaling request should be sent. A destination adjacency is chosen for the signaling message based on various attributes of the message (for example, based on source account or adjacency). Routing policy is applied to new call events and to subscriber registration events.

In releases earlier than Cisco IOS XE Release 3.2S, textual usernames would bypass NA and proceed to route analysis, where they could be matched. From Cisco IOS XE Release 3.2S, NA can validate both dialed digits and textual usernames.

Also, in releases earlier than Cisco IOS XE Release 3.2S, dst-address in NA could be edited, but not src-address. From Cisco IOS XE Release 3.2S, src-address in NA can also be edited. The task of editing src-address can only be performed on digit strings, as in the case of editing dst-address.
In Cisco IOS XE Release 3.2S, `na-src-name-anonymous-table` command was introduced to determine whether the source number's display name or presentation number is anonymous.

Call Admission Control (CAC) limits the number of concurrent calls and registrations, and restricts the media bandwidth dedicated to active calls. It allows for load control on other network elements by rate limiting. Certain events can be completely blocked (using a blacklist) or freely allowed (using a whitelist), based on certain attributes.

Not all policies are mandatory:

- To call between subscribers, only endpoint routing policy is required.
- To call between telephone numbers, only call routing policy is required.
- Number analysis and admission control are optional, although they are likely to be required by the user.

Policies refer to accounts and adjacencies by name. Therefore, you may find it useful to configure and name adjacencies before configuring policies although this is not required.

The following sections describe the many concepts critical to understanding how to implement Cisco Unified Border Element (SP Edition) policies:

- Cisco Unified Border Element (SP Edition) Policies
- Number Analysis Policies
- Routing
- H.323 Call Routing Features
- Call Admission Control

## Cisco Unified Border Element (SP Edition) Policies

This section describes the following Cisco Unified Border Element (SP Edition) policies:

- Policy Events
- Policy Stages
- Policy Sets
- Policy Tables

### Policy Events

Policies are applied to the following events:

- **New calls**—When new SIP or H.323 calls are signaled to the Cisco Unified Border Element (SP Edition), Cisco Unified Border Element (SP Edition) applies a policy to determine what happens to the new call request and what constraints the call must satisfy during its lifetime.
- **Call updates**—If one of the endpoints in a call attempts to renegotiate new media parameters, Cisco Unified Border Element (SP Edition) applies policy to ratify the attempt.
- **Subscriber registrations**—If a subscriber attempts to register through Cisco Unified Border Element (SP Edition), Cisco Unified Border Element (SP Edition) applies policy to determine what happens to the registration request.
Policy Stages

In the context of SIP and H.323 calls, three distinct stages of a policy are applied in a sequence to the policy events. The stages are:

- Inbound number analysis
- Routing
- Outbound number analysis
- Admission control

Some of these policy stages are skipped for particular types of events. Figure 7-2 shows the sequence of the policy stages for each event type.

If the policy stages fail, the call is rejected and the failure is propagated back to the calling device (using either session initiation protocol (SIP) or H.323 signaling, as appropriate) with the error codes in Table 7-2.

<table>
<thead>
<tr>
<th>Component</th>
<th>Resulting SIP Error Code</th>
<th>Resulting H.323 Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number analysis</td>
<td>604 “Does not exist anywhere”</td>
<td>ITU-T Q.931 Release Complete UUIE with H.225 Reason field unreachableDestination</td>
</tr>
<tr>
<td>Routing</td>
<td>604 “Does not exist anywhere”</td>
<td>ITU-T Q.931 Release Complete UUIE with H.225 Reason field unreachableDestination</td>
</tr>
<tr>
<td>Call Admission Control</td>
<td>503 “Service Unavailable”</td>
<td>ITU-T Q.931 Release Complete UUIE with H.225 Reason field noPermission</td>
</tr>
</tbody>
</table>

Note: If the call fails at the routing or Call Admission Control phase, it is released. There is no attempt to retry. Whether or not to retry is left to the upstream (calling) device to decide.

The following sections describe policy stages in more detail:

- Number Analysis
- Routing
- Admission Control
Number Analysis

Number Analysis (NA) determines whether a set of dialed digits or source number represents a valid telephone number. This is achieved by configuring one or more tables of valid source number and dialed digit strings using a limited-form regular-expression syntax, then matching the actual source number or dialed digits against the different strings in the tables.

NA policy is applied only to new call events. If NA determines that a new call does not contain a valid set of source numbers or dialed digits, Cisco Unified Border Element (SP Edition) rejects the call, using the error code described in the Policy Stages section.

NA rules are sensitive to the source account and source adjacency of a call, which allows different dial plans to be configured for different customer organizations, or even for different endpoints.

In addition to validating a source number and dialed number, NA policy can also:

- Reformat the dialed digits into canonical form; for example, E.164 format.
- Label the call with a category, which is used by the later stages of policy.

Routing

Routing determines the next-hop VoIP signaling entity to which a signaling request should be sent. Routing of VoIP signaling messages occurs in two stages:

- Policy-based routing—The first stage of routing. In policy-based routing, a destination adjacency is chosen for the signaling message, based on various attributes of the message, discussed later.

- Protocol-based routing—Takes place after policy-based routing. Protocol-based routing uses a VoIP protocol-specific mechanism to deduce a next-hop IP address from the signaling peer configured for the destination adjacency chosen by policy-based routing.

For example, if the destination adjacency is a SIP adjacency and the signaling peer is uk.globalisp.com, Cisco Unified Border Element (SP Edition) uses domain name server (DNS) or IP lookup to determine the IP address and port of the SIP server for the domain uk.globalisp.com, and forwards the appropriate signaling message to that IP address and port.

Routing policy is applied to new call events and to subscriber registration events.

If a new call event matches an existing subscription, the call is routed automatically to the source IP address and port of the original subscriber registration. No configured policy is required to achieve this, and no configured policy can influence the routing of such calls.

Routing policy is not applied to call update events; call update signaling messages are routed automatically to the destination adjacency that was chosen for the new call event that originated the call.

It is possible that an event cannot be routed, if its attributes do not match a suitable configured routing rule. In such cases, Cisco Unified Border Element (SP Edition) rejects the event using a suitable error code.

Regular expression based routing feature allows the user to configure routing rules that use regular expressions to match the user name or domain part of a source or destination SIP URI.

SBC supports SIP trunk-group ID routing which provides call routing based on the value of the source or destination TGID parameters in the received SIP INVITE message.
Note
A trunk in a network is a communication path connecting two switching systems used in the establishment of an end-to-end connection. A trunk-group is a set of trunks, traffic engineered as a unit, for the establishment of connections within or between switching systems in which all of the paths are interchangeable. TGID is a string that identifies a trunk-group uniquely within a given context.

Admission Control

Call admission control determines whether an event should be granted or refused based on configured limits for network resource utilization. There are two reasons for performing admission control.

- To defend load-sensitive network elements, such as softswitches, against potentially harmful levels of load precipitated by singular events, such as DoS attacks, natural or man-made disasters, or mass-media phone-ins.
- To police the Service Level Agreements (SLAs) between organizations, to ensure that the levels of network utilization defined in the SLA are not exceeded.

Call admission control policy is applied to all event types. If an event is not granted by admission control policy, then Cisco Unified Border Element (SP Edition) rejects it with a suitable error code.

Policy Sets

A policy set is a group of policies that can be active on Cisco Unified Border Element (SP Edition) at any one time. If a policy set is active, then Cisco Unified Border Element (SP Edition) uses the rules defined within it to apply policy to events. You can create multiple policy sets on a single Cisco Unified Border Element (SP Edition).

A policy set has two potential uses:

- It enables you to atomically modify the configured policy by creating a copy of the currently active policy set, making all necessary changes, reviewing the modified policy, and then switching the active policy set. If a problem is discovered with the new policy set after it is activated, Cisco Unified Border Element (SP Edition) can be switched back to using the previous policy set with a single command.
- It enables you to create different policy sets for use at different times and to switch between them at the appropriate times.

Number analysis and routing are configured in a call policy set. Admission control is configured in a CAC policy set.

A new policy set can either be created empty (that is, without any configured policies), or created as a copy of another policy set. A policy set can be deleted, provided that it is not the active policy set.

When the Cisco Unified Border Element (SP Edition) is initialized, there are no active policy sets. At any time after initialization, the active policy set can be undefined. While there is no active routing policy, each event that requires routing is rejected.

From Cisco IOS XE Release 3.2S, the administrative domain allows a user to create separate groups of start indexes for number analysis, route analysis, and a CAC policy that can point to different policy sets. The administrative domain is then attached to the adjacencies for both incoming and outgoing analysis stages.

You can designate an inactive call policy set as the active call policy set at any time. However, you cannot directly modify an active call policy set. To modify an active call policy set, perform the copy-and-swap procedure.
You can designate an inactive CAC policy set as the active CAC policy set at any time. You can also modify an active CAC policy set by adding a new table in the CAC policy set. Note that you can create an entry in an existing table of an active CAC policy set only if the table type is **limit all** or **policy-set**. To perform a modification of this type, you must perform the copy-and-swap procedure.

You can define multiple policy sets that are active and select policy sets that can be used at each call analysis stage based on the adjacency setting. To modify a policy that may be referenced by multiple administrative domains, perform the copy-and-swap procedure.

### Modifying Active CAC Policy Sets

The procedure to modify an active CAC policy set is the same as the procedure to create a CAC policy set. This procedure is described in the "Configuring Call Admission Control Policy Sets, CAC Tables, and Global CAC Policy Sets" section on page 7-115. The difference lies in the checks the system performs at the end of each of these procedures. The newly modified CAC policy set is activated only after it is determined that the following conditions are met by all the CAC policy tables that are reachable from the modified CAC policy table. A failure message is displayed if any of the CAC policy tables do not meet any of these conditions.

- The table is active.
- All table lookup actions in the table point to valid tables.
- None of the table lookup actions result in a CAC configuration loop.
- All table entry values are valid. For example, the scope name or match prefix length must meet the specified criteria.

Note that the modified CAC policy set is applied only to new incoming calls. Calls that were in progress before the modified CAC policy set is made active are not affected when the modified CAC policy set is made active.

### Copy-and-Swap Procedure

To perform a copy and swap procedure, specify the source policy to be copied, and the destination policy to which the source policy is to be copied. The source policy must be an existing policy set, but the destination policy must not be an existing policy set. To protect policies from being overwritten, an error is generated if an attempt is made to copy to an existing policy set.

The old policy can be referenced by different administrative domains, and have multiple indexes within one administrative domain. When the policies are swapped, all the references pertaining to the source policy are replaced with the destination policy. The swap function replaces the default policy and global policy sets, including any policy set referenced in an administrative domain.

The new policy should be set to complete using the `complete` command before all the references to the old policy are replaced.

We recommend that the new policy is exercised globally before all the references to the old policy are replaced.

---

**Note**

An error is generated if the old policy either does not exist or is in an incomplete state.

The following configuration example describes the steps involved in copying and swapping call policy set 2:

```
Router# show run | b call-policy-set 2
call-policy-set 2
description this is call policy 1
```
Step 1  Copy the existing call-policy-set 2 to a new call-policy-set 20:

Step 2  Modify the new call-policy-set with the necessary changes:

Step 3  Set the new call-policy-set 20 to complete:

Step 4  Swap the policies so that references to policy set 2 are replaced with policy set 20. The swap function replaces the default and global policy sets, including any policy set referenced in an administrative domain:

The following configuration example describes the steps involved in copying and swapping an existing CAC policy set 12:

Step 1  Copy the existing cac-policy-set 12 to a new cac-policy-set 22:
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# sbc MySBC
Router(config-sbc)# sbe
Router(config-sbc-sbe)# cac-policy-set copy source 12 destination 22

Step 2  Modify the new cac-policy-set with the necessary changes:
Router(config-sbc-sbe-cacpolicy)# cac-table TAB1
Router(config-sbc-sbe-cacpolicy-cactable)# entry 1
Router(config-sbc-sbe-cacpolicy-cactable-entry)# $max-call-rate-per-scope 100

Step 3  Set the new cac-policy-set 22 to complete:
Router(config-sbc-sbe)# cac-policy-set 22
Router(config-sbc-sbe-cacpolicy)# complete
Router(config-sbc-sbe-cacpolicy)# exit

Step 4  Swap the policies so that references to policy set 12 are replaced with policy set 22:
Router(config-sbc-sbe)# cac-policy-set swap source 12 destination 22

Policy Tables

All policies on the SBE is configured in a set of tables. This section describes the overall structure of the policy tables, as described in the following sections:

- Nomenclature
- Application of Policy
- Policy Selection
- Policy Table Example

Nomenclature

This section defines some terms that we later use when discussing policy tables.

A policy table has the following properties:

- A name that uniquely identifies the table within the scope of a single policy set. Tables in different policy sets may have the same name.
- A type, which defines the criterion that is used to select an entry from the table.
- A collection of table entries.

A policy table entry is a member of a policy table. It has the following properties:

- A value to match on (the match value). The semantics of this value are determined by the table type. No two entries in the same table may have identical match values.
- An optional action to perform on the event, if it matches this entry.
- An optional name of the next table to search for policy, if the event matches this entry.

Application of Policy

The policy tables are searched whenever an event occurs. The policy to be applied to the event is built up as the tables are searched.

The policy sets contain the following properties, which define which policy tables are searched at each stage of the policy calculation. The call policy set contains:

- First NA policy table to process
First routing policy table to process for calls
First routing policy table to process for endpoint registrations

The CAC policy set contains the first admission control policy table.

When an event occurs, the policy tables are searched as follows. This procedure is followed once for every stage of policy to which an event is subjected.

- The first table for the particular stage of the policy calculation is obtained from the active configuration set.
- The type of the table defines which of the event’s attributes (for example, the destination number or the source adjacency) is being examined by this table.
- This attribute is compared against the match value of every entry in the table. This results in either exactly one entry matching the event, or no entries matching the event.
- If an entry matches the event, then the action associated with that entry is performed. After the action is performed, if the entry contains the name of a next table, that table is processed. If there is no next table, then the policy calculation is complete and processing for this stage of policy ends.
- If no entry matches the event, then the policy calculation is complete and processing for this stage of policy ends.

**Policy Selection**

From Cisco IOS Release 3.2S, the SBC can have multiple active configuration sets. However, by using administrative domains, you can select different policy sets for inbound number analysis, routing, CAC, and outbound number analysis for messages based on their source and destination adjacencies.

**Figure 7-3** explains the call processing flow using the policy sets.

The policy set that is to be used for a given administrative domain is defined in the admin-domain mode. Call policy sets specified in the admin-domain mode is given a priority. The priority is required because more than one administrative domain can be specified on an adjacency. The SBC will use the policy-set with the highest priority.

The policy sets must be in a complete state before they are assigned to an administrative domain. A default call-policy-set must be configured before the administrative domain mode is entered. If an inbound NA set, a routing set, or an outbound NA set is undefined, the administrative domain uses the values defined within the default call-policy-set. For more information on administrative domains, see the **Administrative Domains, page 7-59** section.
Information About Implementing Policies

Chapter 7  Implementing Cisco Unified Border Element (SP Edition) Policies

Figure 7-3  Call Processing Flow Using Policy Sets

Call Policy

A signaling event is assigned to the default call policy set if an admin-domain is not specified on the adjacency.

However, you can use different sets of incoming and outgoing number analysis tables based on the administrative domains configured for the incoming and outgoing adjacencies respectively. You can also configure a different routing policy set on a per-adjacency basis.

If more than one administrative domain is associated with the incoming adjacency, the SBC will use the policy set with the highest priority. You should not configure two routing policy sets with the same priority, two inbound NA policy sets with the same priority, or any two outbound NA policy sets with the same priority. The SBC logs an error but uses the policy with the highest index value.

If the adjacencies list any administrative domains that is not listed in the admin-domain mode, they use the priority in the global policy. The SBC logs a configuration warning if an adjacency references an undefined administrative domain.

CAC Policy

All events are limited by the applicable CAC policies indicated by the source and destination administrative domains and the global CAC policy.

The user can configure a CAC policy using different sets of tables based on the administrative domains configured on both the incoming and outgoing adjacencies. It is not required by the administrative domain to specify a CAC policy-set.

Policy Table Example

The following example illustrates the flow of control as policy tables are parsed at a particular stage of policy for a particular event. The event in this example is a new call, received from source account with destination number 129. The stage of policy considered here is routing.
Information About Implementing Policies

This example is provided for illustrative purposes only; routing tables are described in detail in the Routing section. Figure 7-4 shows the relevant routing tables.

Figure 7-4  Policy Table Example

The policy calculation begins by looking up the first policy table to be used by the routing stage. This is the table with name RtgAnalyzeSourceAccount. This table is processed as follows:

- The table type of the table is src-account, so the source account of the new call event is compared with each of the entries in this table.
- The table entry that matches on csi provides a match for this new call event. There is no action associated with this entry, but the entry points to a next table with name RtgAnalyzeDestCSINumber.

The flow of control then passes to the table with name RtgAnalyzeDestCSINumber. This table is processed as follows:

- The cac-scope of the table is dst-number, so the destination number of the new call event is compared with each of the entries in this table.
- The table entry that matches on 1xx provides a match for this new call event. The action associated with this entry is performed; that is, the destination adjacency for the new call event is set to csi-chester.
- This entry does not point to a next table, so the policy calculation for the routing stage ends.

This example shows successful routing of the new call. The outcome is successful because the destination adjacency of the new call is selected before the policy calculation finishes. It is entirely possible for the outcome of routing to be unsuccessful for a new call if the routing policy tables do not
assign a destination adjacency to the call before the routing policy calculation ends. For example, the routing policy illustrated above does not successfully route a new call whose source account is csi and whose destination number is 911.

In this example, a single entry is selected from each table that is traversed during the calculation. In general, at most one entry in any policy table matches an event to which policy is being applied. In cases in which more than one entry would match an event, the best matching entry is selected.

Number Analysis Policies

The following Number Analysis (NA) policies are configured within NA tables and are applied simultaneously to new calls and are described in the following sections:

- Number Validation
- Number Categorization
- Digit Manipulation
- Text Addresses
- Outbound Number Analysis

Number Validation

Number validation is fundamental to the process of traversing number analysis policy tables. A number is validated if the NA tables are traversed and the final entry examined contains an action of accept. A number is not valid if the NA tables are traversed, and the final entry examined contains an action of reject. A number also is not valid if, at any stage of processing the NA tables, a table with no matching entries is encountered.

Number analysis tables can be one of the following types:

- **dst-number**—Tables of this type contain entries whose match values represent complete numbers of Destination. In such tables, an entry matches an event if the entire dialed digit string exactly matches the match value of the entry.
- **dst-prefix**—Tables of this type contain entries whose match values represent number prefixes of Destination. In such tables, an entry matches an event if there exists a subset of the dialed digit string, consisting of consecutive digits taken from the front of the dialed digit string, that exactly matches the match value of the entry.
- **src-number**—Tables of this type contain entries whose match values represent complete numbers of Source. In such tables, an entry matches an event if the entire source digit string exactly matches the match value of the entry.
- **src-prefix**—Tables of this type contain entries whose match values represent number prefixes of Source. In such tables, an entry matches an event if there exists a subset of the source digit string, consisting of consecutive digits taken from the front of the source digit string, that exactly matches the match value of the entry.
- **src-account**—Tables of this type contain entries whose match values are the names of accounts. In such tables, an entry matches an event if the name of the source account of the event exactly matches the match value of the entry.
- **src-adjacency**—Tables of this type contain entries whose match values are the names of adjacencies. In such tables, an entry matches an event if the name of the source adjacency of the event exactly matches the match value of the entry.
carrier-id — Tables of this type contain entries matching the carrier ID.

Digit Matching NA Tables

The format of the match values of entries in NA tables that match on the destination number or destination number prefix is a limited-form, regular expression string representing a string of dialed digits. The syntax used is described in Table 7-1.

<table>
<thead>
<tr>
<th>Syntax Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>Any numerical digit 0 – 9.</td>
</tr>
<tr>
<td>( )</td>
<td>The digit within the parentheses is optional. For example, (0) XXXX represents 0XXXX and XXXX.</td>
</tr>
<tr>
<td>[ ]</td>
<td>One of the digits within the square brackets is used. For example, [01]XXX represents 0XXX and 1XXX. A range of values can be represented within the square brackets. For example, [013-5]XXX represents 0XXX, 1XXX, 3XXX, 4XXX and 5XXX.</td>
</tr>
<tr>
<td>*</td>
<td>The * key on the telephone.</td>
</tr>
<tr>
<td>#</td>
<td>The # key on the telephone.</td>
</tr>
<tr>
<td>.</td>
<td>Digit delimiter</td>
</tr>
<tr>
<td>,</td>
<td>Digit delimiter</td>
</tr>
<tr>
<td>a-f/A-F</td>
<td>Hexadecimal digits</td>
</tr>
</tbody>
</table>

In such tables, it is always possible that more than one entry in the table may match a particular digit string. For example, entries that match 1xx and 12x both match a digit string 129. However, a single entry must be chosen from each table, so the Cisco Unified Border Element (SP Edition) chooses the best matching entry by applying the following rules in the order given.

**Step 1** Choose the longest explicit match.

If the NA table is a dst-prefix type, it is possible that more than one entry specifies an explicit number (that is, one that contains no X characters or [ ] constructs) and matches the dialed number of the event. In this situation, the entry with the longest number has priority.

For example, the dialed number begins 011, the number validation table is a dst-prefix type, and there are two matching entries with numbers 01 and 011. The entry with the number 011 takes priority, because it is a longer number.

**Step 2** If there is no explicit match, choose the longest wildcard match.

If the table does not contain an explicit entry to match the dialed number of the event, the longest wildcard entry that matches takes priority.

**Step 3** If there are multiple wildcard matches of the same length, choose the most explicit where possible.

For example, the dialed number is 01234567890, the NA table is a dst-number type, and there are two matching entries with match values 0123XXXXXXXX and 0123456XXXX. In the first entry, the fifth digit is a wildcard; in the second entry, the eighth digit is a wildcard, so the second entry takes priority.

If the same number is dialed, and a different NA table has matching entries [01]234XXXXXXX and 0XXXXXXXXXX, the second entry takes priority, because in the first entry the first digit is a wildcard.
Number Categorization

Events can be placed into user-defined categories during NA processing. This is achieved by specifying a categorization action in an entry of an NA table. Categories are useful, because they may be referred to later during the admission control policy stage.

At most, one category may be associated with an event. If, during processing of the NA tables, categories are assigned to an event multiple times, then the last category to be assigned is used. When a category is assigned to an event, it cannot be deleted, only replaced with another category.

Digit Manipulation

During number analysis (NA), it is often a requirement to normalize numbers—in other words, convert them from the internal format used by a particular organization or service provider to a canonical format understood globally in the Internet and PSTN.

This is achieved by specifying one or more of the following actions in an entry of an NA table:

- **del-prefix N**—This action removes the leading n digits from the dialed digit string, or deletes the entire string if it is n or fewer digits long.
- **del-suffix n**—This action removes the final n digits from the dialed digit string, or deletes the entire string if it is n or fewer digits long.
- **add-prefix digit string**—This action adds the given digit string to the front of the dialed digit string.
- **replace digit string**—This action replaces the entire dialed digit string with the given digit string.

Text Addresses

From Cisco IOS XE Release 3.2S, NA supports both textual username and digit matching. The table name na-src-number-table was changed to na-src-address-table, na-dst-number-table to na-dst-address-table, and na-dst-number-attr-table to na-carrier-id-table.

To match the text addresses, the existing match number is modified to read the match address. The **match-address** command can include a suffix of digits or regex.

In number analysis, you can define the following matching criteria types:

- Digit matching matches the dialed digit strings using specialized digit regex.
- Regex matching is applicable only to textual usernames, and offers a basic regular expression (BRE) syntax.

**Note**
Comparison of dialed digits and regex is possible. To compare a fixed string, a regex without any regex metacharacters should be used.

Outbound Number Analysis

Outbound Number Analysis allows the configuration of the source and destination numbers from the canonical form to a form that is appropriate for the destination administrative domains. The configuration of Outbound Number Analysis is similar to that of Inbound Number Analysis, which is converted from the source administrative domain form to the canonical form.
Outbound Number Analysis is performed automatically after successful routing. Outbound Number Analysis is processed using the `call-policy-set outbound-na` command in the destination administrative domain.

## Routing

This section describes the following routing policies:
- Routing Tables and Adjacencies
- Number Manipulation
- Hunting
- Regular Expression-Based Routing

### Routing Tables and Adjacencies

This section explains how routing tables are configured on the Cisco Unified Border Element (SP Edition).

The inputs to the policy-based routing stage are as follows:
- The destination number of the event, which is the post-NA dialed digit string (that is, it may have been modified from the original dialed digit string)—This input is present only if the event is a new call.
- The source number of the event—This input is present only if the event is a new call.
- The source adjacency of the event.
- The source account of the event.

The routing policy tables examine some or all of these inputs, and produce one of the following outputs:
- A single destination adjacency.
- A group of adjacencies used for load balancing. One of these is chosen, depending on the load previously sent to the adjacencies in this group.

Routing tables represent one of the following types:
- **dst-address**—Tables of this type contain entries matching the dialed number (after number analysis). These values are either complete numbers or number prefixes (depending on whether the `prefix` parameter is given). Without the `prefix` parameter, an entry matches an event if the dialed digit string exactly matches the match value of the entry. With the `prefix` parameter, an entry matches an event if there exists a subset of the dialed digit string, consisting of consecutive digits taken from the front of the dialed digit string that exactly matches the match value of the entry.

Routing actions also match text user name using a regular expression rather than a literal text string. Routing actions are considered to match if the regular expression matches at least one part of the address.

- **src-address**—Tables of this type contain entries matching the dialer’s number or SIP user name. These values are either complete numbers or number prefixes (depending on whether the `prefix` parameter is given). Without the `prefix` parameter, an entry matches an event if the entire digit string representing the calling number exactly matches the match value of the entry. With the `prefix` parameter, an entry matches an event if there exists a subset of the digit string that represents the calling number, consisting of consecutive digits taken from the front of this string that exactly match the match value of the entry.
Routing actions also match text user name using a regular expression rather than a literal text string. Routing actions are considered to match if the regular expression matches at least one part of the address.

- **src-account**—Tables of this type contain entries matching the names of accounts. In such tables, an entry matches an event if the name of the source account of the event exactly matches the match value of the entry.
- **src-adjacency**—Tables of this type contain entries matching the names of adjacencies. In such tables, an entry matches an event if the name of the source adjacency of the event exactly matches the match value of the entry.
- **src-domain**—Tables of this type contain entries matching the source domain names.
  Routing actions also match domain names using full regular expressions rather than the limited range of regular expression matching. Routing actions are considered to match if the regular expression matches at least one part of the domain.
- **dst-domain**—Tables of this type contain entries matching the destination domain names.
  Routing actions also match domain names using full regular expressions rather than the limited range of regular expression matching. Routing actions are considered to match if the regular expression matches at least one part of the domain.
- **carrier-id**—Tables of this type contain entries matching the carrier ID.
- **round-robin-table**—A group of adjacencies are chosen for an event if an entry in a routing table matches that event and points to a round-robin adjacency table in the next-table action. A round-robin adjacency table is a special type of policy table, whose events do not have any match-value parameters, nor next-table actions. Its actions are restricted to setting the destination adjacency and performing digit manipulation.
- **category**—Tables of this type contain entries matching on the category that was assigned to the call during number analysis. You assign the category during number analysis.
- **time**—Tables of this type contain entries matching on a user-configured time. The entries can have overlapping match periods. Time periods can be specified by year, month, date, day of the week, hour, or minute.
- **least-cost**—Tables of this type contain entries matching on the user-configured precedence (cost) of the entries. If more than one entry has an equal cost, an entry is selected based on a user-configured weight or an entry is selected based on the number of active calls on each route. If routing fails, then the adjacency with the next lowest cost is selected.
- **src-trunk-group-id**—Tables of this type contain entries matching the source TGID or TGID context parameters and action type to perform the call routing.
- **dst-trunk-group-id**—Tables of this type contain entries matching the destination TGID or TGID context parameters and action type to perform the call routing.

The rules specified in the "Digit Matching NA Tables? section on page 7-15 govern the format and matching rules of the match-values of the entries in routing tables of type dst-number, dst-prefix, src-number and src-prefix.

**Number Manipulation**

The number manipulation feature enables you to specify various number manipulations that can be performed on a dialed number after a destination adjacency has been selected. Number manipulation can be configured as a routing policy.
This enhancement affects the billing functionality as it allows the Cisco Unified Border Element (SP Edition) to display both the original and the edited dialed number for a call. For example:

```xml
<party type="orig" phone="01234567890"/>
<party type="term" phone="2345678931" editphone="11111111111"/>
```

Note

The phone numbers in the above example are not real.

The number manipulation feature requires that the edit action be allowed in the routing policy entries. The edit action takes the same parameters as the edit action for the number analysis tables, enabling you to delete a number of characters from the beginning or end of the dialed string, add digits to the start of the string, or replace the entire string with another. For example, if the following table were matched:

```plaintext
call-policy-set 1
rtg-src-adjacency-table table1
  entry 1
    match SipAdj1
    edit del-prefix 3
    dst-adjacency SipAdj2
    action complete
  end
end
```

then the dialed string would have the first three of its digits deleted.

In the number analysis stage you can specify categories as shown below.

```plaintext
call-policy-set 1
  first-inbound-na-table check-accounts
  na-src-account-table check_accounts
  entry 1
    match-account hotel_foo
    action next-table hotel_dialing_plan
  entry 2
    match-account hotel_bar
    action next-table hotel_dialing_plan
  entry 3
    match-account internal
    action accept
  na-dst-prefix-table hotel_dialing_plan
  entry 1
    match-prefix XXX
    category internal
    action accept
  entry 2
    match-prefix 9XXX
    category external
    action accept
```

Later during routing, the calls are routed based on assigned categories.

```plaintext
call-policy-set 1
  first-call-routing-table start_routing
  rtg-category-table start_routing
  entry 1
    match-category internal
    action next-table internal_routing
  entry 2
    match-category external
    action next-table external_routing
  rtg-src-adjacency-table internal_routing
  entry 1
    match-adjacency sip_from_foo
```

```xml
<party type="orig" phone="01234567890"/>
<party type="term" phone="2345678931" editphone="11111111111"/>
```
Information About Implementing Policies

You can also specify various number manipulations to be performed on a dialing or dialed number after a destination adjacency is selected.

The following example adds a prefix of “123” to the source number, for all calls coming in on “SipAdj1” adjacency and destined to “SipAdj2”.

```
call-policy-set 1
  rtg-src-adjacency-table table1
  entry 1
    match SipAdj1
    edit-src add-prefix 123
    dst-adjacency SipAdj2
    action complete
```

Hunting

Cisco Unified Border Element (SP Edition) can hunt for other routes or destination adjacencies in case of a failure. Hunting means the route is retried. Cisco Unified Border Element (SP Edition) supports hunting of SIP and H.323 calls. Hunting can be configured as a routing policy.

There are several ways in which failures can occur, including the following:

- **CAC policy refusing to admit a call**
  
  If a CAC policy rejects a call, the SBC automatically attempts to reroute the call using the Routing Policy Service (RPS). RPS decides where to route onward signaling requests by using the configured policy in the RPS. The call is then tested against CAC policy again.

- **Routing Policy Services being unable to route a call**

- **Call setup failure being received from SIP or H.323.**
  
  When the SBC receives a call setup failure notification from H.323 or SIP, it is notified whether or not it should attempt to reroute the call, depending upon the error code.
If an SIP or H.323 adjacency attempts to route a call, and the attempt fails, it receives an error code. You can configure which error codes trigger hunting or rerouting.

- If the error code received by the adjacency matches an entry on this list, RPS is signalled to reroute the call. Rerouting then occurs unless the number of attempts exceeds the limit set as the maximum number of routing attempts that SBC makes. The default is three attempts.
- If the error code received by the adjacency does not match an entry on this list, RPS is signalled not to reroute the call.

For both SIP and H.323 call, you can configure a list of error codes or failure return codes to trigger hunting or rerouting for a particular adjacency by using the `sip hunting-trigger error-codes` or `hunting-trigger error-codes` commands.

You can also configure a list of H.323 error codes at a global level, by using the `hunting-trigger` command in the global H.323 configuration mode.

*SIP error codes* are numeric error codes. H.323 error codes are textual. See the table.

Hunting finishes when one of the following conditions is met:

- The call is successfully routed.
- The SBC receives a call setup failure notification with the instruction not to continue hunting, in which case the call fails.
- The SBC has made the number of specified routing attempts and the call has not been successfully routed, in which case the call fails.
- The SBC has tried all available adjacencies, and the call has not been successfully routed, in which case the call fails.

H.323 hunting has the additional hunting modes of alternate endpoints and multiARQ hunting. See the section on page 7-24.

For information on configuring SIP and H.323 hunting, see the section on page 7-107.
Table 7-2 lists the supported error codes that you can configure to trigger hunting of SIP or H.323 calls.

<table>
<thead>
<tr>
<th>Supported SIP Error Codes</th>
<th>Supported H.323 Error Codes</th>
</tr>
</thead>
<tbody>
<tr>
<td>400 - Bad Request</td>
<td>unreachableDestination</td>
</tr>
<tr>
<td>401 - Unauthorized</td>
<td>noPermission</td>
</tr>
<tr>
<td>402 - Payment Required</td>
<td>noBandwidth</td>
</tr>
<tr>
<td>403 - Forbidden</td>
<td>destinationRejection</td>
</tr>
<tr>
<td>404 - Not Found</td>
<td>gatewayResources</td>
</tr>
<tr>
<td>405 - Method Not Allowed</td>
<td>badFormatAddress</td>
</tr>
<tr>
<td>406 - Not Acceptable</td>
<td>securityDenied</td>
</tr>
<tr>
<td>407 - Proxy Authentication Required</td>
<td>the internally-defined value “connectFailed”</td>
</tr>
<tr>
<td>408 - Request Timeout</td>
<td>—</td>
</tr>
<tr>
<td>409 - Conflict</td>
<td>—</td>
</tr>
<tr>
<td>410 - Gone</td>
<td>—</td>
</tr>
<tr>
<td>411 - Length Required</td>
<td>—</td>
</tr>
<tr>
<td>413 - Request Entity Too Large</td>
<td>—</td>
</tr>
<tr>
<td>414 - Request URI Too Long</td>
<td>—</td>
</tr>
<tr>
<td>415 - Unsupported Media Type</td>
<td>—</td>
</tr>
<tr>
<td>416 - Unsupported URI Scheme</td>
<td>—</td>
</tr>
<tr>
<td>420 - Bad Extension</td>
<td>—</td>
</tr>
<tr>
<td>421 - Extension Required</td>
<td>—</td>
</tr>
<tr>
<td>423 - Interval Too Brief</td>
<td>—</td>
</tr>
<tr>
<td>480 - Temporarily Unavailable</td>
<td>—</td>
</tr>
<tr>
<td>481 - Call/Transaction Does Not Exist</td>
<td>—</td>
</tr>
<tr>
<td>482 - Loop Detected</td>
<td>—</td>
</tr>
<tr>
<td>483 - Too Many Hops</td>
<td>—</td>
</tr>
<tr>
<td>484 - Address Incomplete</td>
<td>—</td>
</tr>
<tr>
<td>485 - Ambiguous</td>
<td>—</td>
</tr>
<tr>
<td>486 - Busy Here</td>
<td>—</td>
</tr>
<tr>
<td>487 - Request Terminated</td>
<td>—</td>
</tr>
<tr>
<td>488 - Not Acceptable Here</td>
<td>—</td>
</tr>
<tr>
<td>491 - Request Pending</td>
<td>—</td>
</tr>
<tr>
<td>493 - Undecipherable</td>
<td>—</td>
</tr>
<tr>
<td>500 - Server Internal Error</td>
<td>—</td>
</tr>
<tr>
<td>501 - Not Implemented</td>
<td>—</td>
</tr>
<tr>
<td>502 - Bad Gateway</td>
<td>—</td>
</tr>
<tr>
<td>503 - Service Unavailable</td>
<td>—</td>
</tr>
<tr>
<td>504 - Server Time-Out</td>
<td>—</td>
</tr>
</tbody>
</table>
Regular Expression-Based Routing

Regular expression based routing allows the user to configure routing rules that use regular expressions to match the user name or domain part of a source or destination SIP URI.

Routing actions match text user name using a regular expression rather than a literal text string when “regex” keyword is used. Routing actions are considered to match if the regular expression matches at least one part of the address.

Table 7-3 shows the basic regular expression (BRE) implementation for the supported regex characters.

Table 7-3 BRE Implementation

<table>
<thead>
<tr>
<th>Metacharacter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.</td>
<td>Matches any single character. Within POSIX bracket expressions, the dot character matches a literal dot. For example, a.c matches &quot;abc&quot;, etc., but [a.c] matches only &quot;a&quot;, &quot;,&quot;, or &quot;c&quot;.</td>
</tr>
<tr>
<td>[ ]</td>
<td>A bracket expression. Matches a single character that is contained within the brackets. For example, [abc] matches &quot;a&quot;, &quot;b&quot;, or &quot;c&quot;. [a-z] specifies a range which matches any lowercase letter from &quot;a&quot; to &quot;z&quot;. The - character is treated as a literal character if it is the last or the first character within the brackets, or if it is escaped with a backslash: [abc-], [-abc], or [a-bc].</td>
</tr>
<tr>
<td>^</td>
<td>Matches a single character that is not contained within the brackets. For example, [^abc] matches any character other than &quot;a&quot;, &quot;b&quot;, or &quot;c&quot;. [^a-z] matches any single character that is not a lowercase letter from &quot;a&quot; to &quot;z&quot;. As above, literal characters and ranges can be mixed.</td>
</tr>
<tr>
<td>^</td>
<td>Matches the starting position of the string.</td>
</tr>
<tr>
<td>$</td>
<td>Matches the ending position of the string.</td>
</tr>
<tr>
<td>( )</td>
<td>Defines a marked subexpression. The string matched within the parentheses can be recalled later (see the next entry, \n).</td>
</tr>
<tr>
<td>\n</td>
<td>Defines what the nth marked subexpression matched, where n is a digit from 1 to 9. This construct is theoretically irregular and was not adopted in the POSIX ERE syntax. Some tools allow referencing more than nine capturing groups.</td>
</tr>
<tr>
<td>*</td>
<td>Matches the preceding element zero or more times.</td>
</tr>
<tr>
<td>{m,n}</td>
<td>Matches the preceding element at least m and not more than n times. For example, a{3,5} matches only &quot;aaa&quot;, &quot;aaaa&quot;, and &quot;aaaaa&quot;.</td>
</tr>
</tbody>
</table>
The rtg-src-address and rtg-dst-address tables contain entries matching the dialed number (after number analysis). At run-time, when the Request-URI is processed, the username is parsed to determine if the username is considered to be “textual” or “dialed-digits”. It is initially assumed that the username is a dialed-digit string, and the username will considered to be textual only if non-dialed digit characters are encountered. Having determined this type, only policy entries matching this type are evaluated.

When configuring policy entries which match on rtg-src-address or rtg-dst-address table, it is important to configure the match-address correctly to ensure the policy entry is evaluated. In order to assist in configuration, the type of match address will be assessed and configured automatically if not specifically configured.

You can configure one of the following three choices explicitly:

match-address address [digits] (limited digit string regex)
match-address address [string] (string (textual) comparison on textual username only)
match-address address [regex] (regular expression on string (textual) usernames only)

Example:

Valid entries:
match-address (0)1234[56] digits
match-address username string
match-address [Uu]sername regex

Invalid entries:
match-address 1234 string (cannot perform a string match on dialed digits)
match-address 1234 regex (cannot perform a regex match on dialed-digits)
match-address [abc] regex (abc are valid dialed digits and #, * and d are also valid dialed digits)

In this case the entry is evaluated at configuration time and error responses generated if there is a perceived mismatch in the type and match-address.

**H.323 Call Routing Features**

In addition to the features described in the [Routing? section on page 7-17](#) that also apply to H.323 calls, Cisco Unified Border Element (SP Edition) supports various H.323-specific call routing features.

The H.323 call routing features are:

- **H.323 Hunting, page 7-25**
- **Picking a Next Hop in Routing Policy, page 7-26**
- **Support for H.323 addressing, page 7-26**
- **DNS Name Resolution, page 7-26**
- **Number Validation and Editing, page 7-26**
- **Load Balancing, page 7-27**
- **Inter-VPN Calling, page 7-27**
H.323 Hunting

Cisco Unified Border Element (SP Edition) supports hunting of H.323 calls. Cisco Unified Border Element (SP Edition) hunts for other routes or destination adjacencies in the event of a failure. Hunting re-routes the call in response to a specific user-configured event or error code.

H.323 hunting or re-routing operates in the following ways based on whether the adjacency is a gatekeeper or non-gatekeeper adjacency:

- For a gatekeeper adjacency, the SBC can cycle through a list of potential signaling next hops based on input from the gatekeeper. Alternate Endpoints and MultiARQ are two methods that allow the gatekeeper to provide the SBC with this list.

  If H.323 has a list of alternate endpoints for a call, H.323 tries each of these in turn before reporting a routing failure to the RPS.

  MultiARQ is described in the "MultiARQ Hunting" section.

- For a non-gatekeeper adjacency, or where all the next hops on a gatekeeper adjacency have been exhausted, the SBC can re-route the call to a different adjacency in the "hunt group" (specifically, the round-robin-table or least-cost routing table). For more information on routing tables, see the "Routing Tables and Adjacencies" section on page 7-17.

MultiARQ Hunting

Cisco Unified Border Element (SP Edition) supports a non-standard H.323 mechanism for hunting for other routes or destination adjacencies. This is based on issuing multiple Admission Requests (ARQs) to a Gatekeeper for a single call.

The SBC sends an ARQ (Admission Request) when an incoming call is received on a gatekeeper adjacency, or an outgoing call needs to be made on a gatekeeper adjacency. For an outgoing call, the gatekeeper returns the signaling address of the endpoint that the SBC should contact.

MultiARQ hunting occurs under the following circumstances:

- The H.323 endpoint sends an ARQ to a Gatekeeper as part of establishing an outbound call leg.
- The Gatekeeper contacts other network entities and identifies one or more potential endpoints.
- The Gatekeeper returns an admissionConfirm (ACF) containing a single destinationInfo and no alternateEndpoints.
- The H.323 endpoint attempts to contact the endpoint identified in the ACF. The endpoint either rejects the call or is unreachable.

The MultiARQ hunting continues until one of the following conditions is met:

- An endpoint is contacted and the call completes.
- A Gatekeeper ARQ retry is required, but the hard-coded limit on the number of permitted retry ARQs has been reached. This number is a customizable constant in h323cust.h, and is currently set to 32.
- The Gatekeeper returns an admissionReject, indicating that there are no further suitable endpoint identifiers.
- An endpoint returns a rejectReason which is not configured as a hunting trigger.
- An endpoint cannot be contacted, and connectFailed is not configured as a hunting trigger.

For information on configuring MultiARQ Hunting, see the "Configuring H.323 MultiARQ Hunting" section on page 7-112.
Picking a Next Hop in Routing Policy

When receiving an incoming H.323 call, Cisco Unified Border Element (SP Edition) carries out routing to determine the next hop for the call.

SBC policy allows calls to be routed to one of the following:

- signaling peer (such as a gateway)
- outgoing gatekeeper

When a gatekeeper is used, the gatekeeper is responsible for resolving the called party number to a next hop address.

In a SBC configuration, a routing next hop is identified by an adjacency name. The adjacency is configured with the address of the next hop gateway or gatekeeper.

Support for H.323 addressing

All H.323 calls through Cisco Unified Border Element (SP Edition) need to specify a called party number. A called party number may optionally be supplied in the Q.931 calledPartyNumber or the H.225 destinationAddress, with the former taking priority. If a called party number is not present in either of these fields, then the SBC rejects the call.

Finally, the connected number may also optionally be supplied in the Q.931 connectedNumber or the H.225 connectedAddress, with the former taking priority. The connected number indicates the party the call ends up connecting with because during call setup, the call might be redirected or the called number might be edited along the way.

When an H.323 endpoint sends out a Q.931/H.225 message, the called and calling numbers are always placed in the Q.931 fields, not the H.225 fields.

DNS Name Resolution

Domain name server (DNS) name resolution enables you to use the domain name instead of the IP address in an adjacency configuration. You can configure both gatekeeper and non-gatekeeper adjacencies with DNS names.

If you use a DNS name in an adjacency configuration, the name is resolved each time a call is routed out over that adjacency. This process allows DNS-based load-balancing.

Number Validation and Editing

Cisco Unified Border Element (SP Edition) allows validation, editing and categorization of the called and calling party number through a Number Validation configuration.

This can be used for comparing or editing of source or destination telephone numbers or textual usernames. This process is called Number Analysis (NA). Number Analysis (NA) determines whether a set of source or destination digits, or source or destination textual addresses represents a valid address (based on number validation, number categorization, and/or digit manipulation). This is achieved by configuring one or more tables of valid addresses and editing rules in the tables. Matching for digit strings uses a limited-form of specialized regular-expression syntax and matching for textual addresses is done on the basis of the Basic Regular Expression syntax. In both cases, either the entire address or part of the address can be matched.

NA can be optionally configured as a step within the call policy set.
For more information, see the "Number Analysis Policies" section on page 7-14 section and the "Number Analysis" section on page 7-6 in the Implementing Cisco Unified Border Element (SP Edition) Policies chapter.

### Load Balancing

Cisco Unified Border Element (SP Edition) can load balance between H.323 adjacencies using Round Robin or Least Cost Routing configurations.

Round Robin load balancing distributes calls evenly between adjacencies. Least Cost load balancing assigns a priority to each adjacency.

For example, routing might route two consecutive calls onto two different adjacencies.

- For gatekeeper adjacencies, the calls will be admitted on two different gatekeepers. It is up to the gatekeeper routing configuration to determine whether the signaling next hop for each call is the same.
- For non-gatekeeper adjacencies, the signaling next hop will be set to two different gateways (or terminals).

If a gatekeeper adjacency loses contact with the gatekeeper, it is temporarily taken out of service - meaning that the SBC will not attempt to route new calls through it. If there is an alternative route, call setup will continue on the alternative route. You can also manually deactivate an adjacency, which has the same effect.

### Inter-VPN Calling

Cisco Unified Border Element (SP Edition) can peer with H.323 devices in different VPNs simultaneously.

You configure VPNs on a per-adjacency basis. Therefore, inter-VPN calling is simply a matter of your configuring a routing policy that routes calls between adjacencies in different VPNs.

### Call Admission Control

This section describes the following:

- Call Admission Control Overview, page 7-28
- Compound Scopes, page 7-28
- Policy Scopes, page 7-29
- Policy Set Tables and Limit Tables, page 7-32
- Limit Tables, page 7-32
- CAC Table Entry Configuration Commands, page 7-33
- Media Line Removal, page 7-38
- Multiple SBC Media Bypass, page 7-39
- Common IP Address Media Bypass, page 7-43
- CAC Rate Limiting, page 7-45
- Multiple CAC Averaging Periods, page 7-46
- Subscriber Policy, page 7-46
Call Admission Control Overview

Call Admission Control (CAC) allows you to configure policy for accepting or rejecting calls. It allows you to apply detailed policies to certain call options to limit the number of concurrent calls and registrations. CAC can restrict the media bandwidth dedicated to active calls. It allows for load control on other network elements by rate limiting. Certain events can be completely blocked (using a blacklist) or freely allowed (using a whitelist), based on certain attributes.

CAC determines whether an event should be granted or refused based on configured limits for network resource utilization. There are two reasons for performing call admission control:

- To defend load-sensitive network elements, such as softswitches, against potentially harmful levels of load precipitated by singular events, such as DoS attacks, natural or man-made disasters, or mass-media phone-ins.
- To police the Service Level Agreements (SLAs) between organizations, to ensure that the levels of network utilization defined in the SLA are not exceeded.

Call admission control is the final stage of the call policy, so it is applied after number analysis and routing policy. CAC policy is applied to all event types, such as new calls, subscriber registrations, and call updates. If an event is not granted by the CAC policy, then Cisco Unified Border Element (SP Edition) rejects it with a suitable error code.

A CAC policy consists of the following:

- A limit or limits that must not be exceeded.
  Limits, for example, can be set on the maximum number of concurrent calls, the maximum rate of calls, or the maximum bandwidth consumed by calls.
- A scope at which the limits are applied.
  This can be global, per-account, per-adjacency, or any of the scopes defined in Policy scopes. Combinations of scopes can also be used, such as “per account, per number category.” Scope is part of the policy itself. For example, in the policy “maximum 20Kb per call,” the scope is “per call.”

To define an admission control policy, you must define the limit and the scope at which it is applied. For example, you can define a policy such that not more than 10 concurrent calls (limit) could ever be made from a single account (scope).

Although the scope and limits define the policy, they do not determine when the policy is applied. For example, you cannot name a particular account, such as “account1,” as the scope for your policy. Instead, the table-type and match value are used to determine when a policy is applied. Setting “account” as the table-type and “account1” as the match value matches call events from account1.

Compound Scopes

Compound scopes provide a more elaborate set of options for configuring policy. Certain policy scopes can be combined to create compound scopes. To combine scopes, configure each scope using a separate first-cac-scope or cac-scope command.

The following are examples of compound scopes:

- If you want to restrict the number of calls between any pair of adjacencies to 20, you could create a policy with MaxCalls = 20 and a scope of “src_adjacency, dst_adjacency.” This policy would restrict the number of calls between any pair of adjacencies to 20. However, it would not limit the total number of calls out of any adjacency, nor the total number of calls into any adjacency.
You can define an admission control policy at a compound scope of “source adjacency and category,” and set the maximum concurrent calls in this scope to 10. This policy would restrict the number of concurrent calls of the same category that each adjacency could make to 10. The scope field value is src-adjacency, category.

### Policy Scopes

Table 7-4 defines the scopes in which call admission policies can be applied and specifies whether each of these scopes can be combined with other scopes.

<table>
<thead>
<tr>
<th>Scope Option or Value of Scope Field</th>
<th>Scope</th>
<th>Description</th>
<th>Can Scope Be Combined?</th>
</tr>
</thead>
<tbody>
<tr>
<td>account</td>
<td>Per account</td>
<td>The limits specified in this scope apply to all the events from the same account.</td>
<td>Yes, except the dst-account and src-account scopes</td>
</tr>
<tr>
<td>adjacency</td>
<td>Per adjacency</td>
<td>The limits specified in this scope apply to all the events from the same adjacency.</td>
<td>Yes, except the src-adjacency, dst-adjacency, src-adj-group, and dst-adj-group scopes</td>
</tr>
<tr>
<td>adj-group</td>
<td>Per adjacency group</td>
<td>The limits specified in this scope apply to all events sent to or received from the same adjacency group. For example, you can restrict the total number of concurrent calls that can be sent to or received from the adjacencies in a single adjacency group by configuring limits in this scope.</td>
<td>Yes, except the adjacency, src-adj-group, and dst-adj-group scopes</td>
</tr>
<tr>
<td>call</td>
<td>Per call</td>
<td>The limits specified in this scope apply to any single call. For example, you can restrict the per-call bandwidth or the allowed call update rate by configuring limits in this scope. Note that some limits are invalid in this scope.</td>
<td>No</td>
</tr>
<tr>
<td>category</td>
<td>Per category</td>
<td>The limits specified in this scope apply to all events that have been placed in the same category by the number analysis policy tables. For example, you can restrict the total number of concurrent calls in any single category by configuring limits in this scope.</td>
<td>Yes</td>
</tr>
<tr>
<td>dst-account</td>
<td>Per destination account</td>
<td>The limits specified in this scope apply to all events sent to the same account. For example, you can restrict the total number of concurrent calls that can be sent to any single account by configuring limits in this scope.</td>
<td>Yes, except the account scope</td>
</tr>
</tbody>
</table>
### Table 7-4 Policy Scope Definitions (continued)

<table>
<thead>
<tr>
<th>Scope Option or Value of Scope Field</th>
<th>Scope</th>
<th>Description</th>
<th>Can Scope Be Combined?</th>
</tr>
</thead>
<tbody>
<tr>
<td>dst-adj-group</td>
<td>Per destination</td>
<td>The limits specified in this scope apply to all events sent to the same adjacency group. For example, you can restrict the total number of concurrent calls that can be sent to the adjacencies in a single adjacency group by configuring limits in this scope.</td>
<td>Yes, except the adj-group scope</td>
</tr>
<tr>
<td></td>
<td>adjacency group</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dst-adjacency</td>
<td>Per destination</td>
<td>The limits specified in this scope apply to all events sent to the same adjacency. For example, you can restrict the total number of concurrent calls that can be sent to any single adjacency by configuring limits in this scope.</td>
<td>Yes, except the adjacency scope</td>
</tr>
<tr>
<td></td>
<td>adjacency</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dst-number</td>
<td>Per dialed number</td>
<td>The limits specified in this scope apply to all events that have the same destination number. For example, you can restrict the total number of concurrent calls to any single valid number by configuring limits in this scope.</td>
<td>Yes</td>
</tr>
<tr>
<td>global</td>
<td>Global</td>
<td>The limits specified in this scope apply to SBC as a whole.</td>
<td>No</td>
</tr>
<tr>
<td>src-account</td>
<td>Per source account</td>
<td>The limits specified in this scope apply to all events received from the same account. For example, you can restrict the total number of concurrent calls that can be initiated from any single account by configuring limits in this scope.</td>
<td>Yes, except the account scope</td>
</tr>
<tr>
<td>src-adj-group</td>
<td>Per source adjacency</td>
<td>The limits specified in this scope apply to all events received from the same adjacency group. For example, you can restrict the total number of concurrent calls that can be initiated from the adjacencies in a single adjacency group by configuring limits in this scope.</td>
<td>Yes, except the adjacency and adj-group scopes</td>
</tr>
<tr>
<td></td>
<td>adjacency group</td>
<td></td>
<td></td>
</tr>
<tr>
<td>src-adjacency</td>
<td>Per source adjacency</td>
<td>The limits specified in this scope apply to all events received from the same adjacency. For example, you can restrict the total number of concurrent calls that can be initiated from any single adjacency by configuring limits in this scope.</td>
<td>Yes, except the adjacency scope</td>
</tr>
<tr>
<td>src-number</td>
<td>Per dialing number</td>
<td>The limits specified in this scope apply to all events that have the same source number. For example, you can restrict the total number of concurrent calls from every single source number by configuring limits in this scope.</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Note

If you are supporting Aggregate Registrations in a non-IMS network, all of the phones behind a device (such as a PBX) are counted as the same subscriber if you are using a per-subscriber scope.

Non-Subscriber Group

When a subscriber scope is enabled, the SBC includes an additional group of ALL “non-subscribers.” The non-subscribers are counted within a special group of the subscriber scope. The non-subscriber group is matched if the call is from a non-subscriber. Limits set in the subscriber scope apply to this non-subscriber group.
Information About Implementing Policies

A “subscriber” is identified using the Address-of-Record that is registered with the registrar. A “subscriber category” is based on the source IP address of the SIP message. When some subscribers sit behind a Network Address Translation (NAT) device and share the same IP address, they are in the same subscriber category. However, they differ among each other by their AOR.

Policy Set Tables and Limit Tables

Call admission control policies are configured using a combination of Policy Set and Limit tables.

A Policy Set table type is applied to all entries defined within the CAC table. Each entry within the table configures its own scope. Every entry in a Policy Set table automatically matches every event that reaches that table. Policy Set tables create multiple policies for each event.

A Limit table type selects the single best matching match value defined in a CAC entry. The scope for the limit table type is inherited from the limit table’s parent table. The entries in a Limit table specify the values to match against and the limits to apply if a match is achieved.

The major difference between a Policy Set table and a Limit table is that the Policy Set table creates multiple policies for a given event, while a Limit table only defines one policy for a given event.

For information on table-types, match values, and when an event matches an entry for Limit Table, see Table 7-5. For information on scope name, scope definition, and whether a scope can be combined, see Table 7-4.

Limit Tables

Table 7-5 lists the types of Limit tables. For each table type, the corresponding Match value is listed, with the conditions under which a match is achieved. If a match is achieved, the corresponding policy is applied to the event.

<table>
<thead>
<tr>
<th>Table Type</th>
<th>Match Value</th>
<th>Conditions Where an Event Matches an Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>account</td>
<td>account name</td>
<td>Match value is the source and/or destination account name.</td>
</tr>
<tr>
<td>adj-group</td>
<td>adjacency group name</td>
<td>Match value is the source and/or destination adjacency group name.</td>
</tr>
<tr>
<td>adjacency</td>
<td>adjacency name</td>
<td>Match value is the source and/or destination adjacency name.</td>
</tr>
<tr>
<td>all</td>
<td>NA</td>
<td>All events match entry</td>
</tr>
<tr>
<td>call-priority</td>
<td>SBC priority</td>
<td>SBC priority is the event call-priority.</td>
</tr>
<tr>
<td>category</td>
<td>category name (assigned during number analysis)</td>
<td>Event has been assigned a category, and match value is the name of the category assigned.</td>
</tr>
<tr>
<td>dst-account</td>
<td>account name</td>
<td>Match value is the destination account name.</td>
</tr>
<tr>
<td>dst-adj-group</td>
<td>adjacency group name</td>
<td>Match value is the destination adjacency group name.</td>
</tr>
<tr>
<td>dst-adjacency</td>
<td>adjacency name</td>
<td>Match value is the destination adjacency name.</td>
</tr>
</tbody>
</table>
Information About Implementing Policies

CAC Table Entry Configuration Commands

Each CAC table consists of a collection of table entries, defined within the CAC table submode. For Policy Set table types, the CAC scope is defined within each entry. If unspecified, the scope defaults to global for that entry.

For Limit table types, the CAC entry specifies a value to match against. The semantics of this match-value are determined by the type of Limit table.

For both table types, the limits defined within the entry are calculated using per scope values. Some limits are not applicable at all scopes. Policy Set table types define the scope within the entry, thus both the limit and the scope are per entry. If you want per entry limits for a Limit table type, then configure the Limit table type to match the scope.


Table 7-6 shows a list of various limits and options that can be configured on an entry in a CAC policy-set table. These configurable command options can be displayed with the following commands:

Router(config-sbc-sbe-cacpolicy-cactable-entry)# cac-table 4
Router(config-sbc-sbe-cacpolicy-cactable)# table-type policy-set
Router(config-sbc-sbe-cacpolicy-cactable)# entry 1
Router(config-sbc-sbe-cacpolicy-cactable-entry)# ?
### Table 7-6 CAC Table Entry Configurable Command Options

<table>
<thead>
<tr>
<th>Configurable Command Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cac-scope</td>
<td>Scope at which CAC limits are applied within each entry in a Policy Set table.</td>
</tr>
<tr>
<td>callee</td>
<td>Callee settings</td>
</tr>
<tr>
<td>callee-codec-list</td>
<td>List of codecs which the callee leg of a call is allowed to use</td>
</tr>
<tr>
<td>callee-hold-setting</td>
<td>The callee hold setting supported</td>
</tr>
<tr>
<td>callee-inbound-policy</td>
<td>Set callee inbound Session Description Protocol (SDP) policy table</td>
</tr>
<tr>
<td>callee-outbound-policy</td>
<td>Set callee outbound SDP policy table</td>
</tr>
<tr>
<td>callee-privacy</td>
<td>The level of privacy processing</td>
</tr>
<tr>
<td>callee-sig-qos-profile</td>
<td>QoS profile to use for callee signalling</td>
</tr>
<tr>
<td>callee-video-qos-profile</td>
<td>QoS profile to use for callee video media</td>
</tr>
<tr>
<td>callee-voice-qos-profile</td>
<td>QoS profile to use for callee voice media</td>
</tr>
<tr>
<td>caller</td>
<td>Caller settings</td>
</tr>
<tr>
<td>caller-codec-list</td>
<td>List of codecs which the caller leg of a call is allowed to use</td>
</tr>
<tr>
<td>caller-hold-setting</td>
<td>The caller hold setting supported</td>
</tr>
<tr>
<td>caller-inbound-policy</td>
<td>Set caller inbound SDP policy table</td>
</tr>
<tr>
<td>caller-outbound-policy</td>
<td>Set caller outbound SDP policy table</td>
</tr>
<tr>
<td>caller-privacy</td>
<td>The level of privacy processing</td>
</tr>
<tr>
<td>caller-sig-qos-profile</td>
<td>QoS profile to use for caller signalling</td>
</tr>
<tr>
<td>caller-video-qos-profile</td>
<td>QoS profile to use for caller video media</td>
</tr>
<tr>
<td>caller-voice-qos-profile</td>
<td>QoS profile to use for caller voice media</td>
</tr>
<tr>
<td>codec-restrict-to-list</td>
<td>Restrict to using codecs from a configured codec list</td>
</tr>
<tr>
<td>early-media-deny</td>
<td>Do not allow early-media</td>
</tr>
<tr>
<td>early-media-timeout</td>
<td>Duration for which to allow early media</td>
</tr>
<tr>
<td>early-media-type</td>
<td>Directions in which to allow early media</td>
</tr>
<tr>
<td>match-value</td>
<td>Match-value of an entry in a CAC Limit table</td>
</tr>
<tr>
<td>max-bandwidth</td>
<td>Maximum bandwidth</td>
</tr>
<tr>
<td>max-call-rate-per-scope</td>
<td>Maximum call rate</td>
</tr>
<tr>
<td>max-channels</td>
<td>Maximum number of channels</td>
</tr>
<tr>
<td>max-in-call-msg-rate</td>
<td>Configure maximum rate of in-call messages. See description of in-call messages in the CAC Rate Limiting section on page 7-45.</td>
</tr>
<tr>
<td>max-num-calls</td>
<td>Maximum number of calls</td>
</tr>
</tbody>
</table>
Nonlimiting CAC Options

CAC allows you to configure policy for accepting or rejecting calls based on limit options such as max-num-calls and max-bandwidth. The CAC scope is used when policing limit options. CAC also allows you to apply a property to a call (rather than a limitation) with nonlimiting options, such as caller-inbound-policy. Scopes have no meaning for nonlimiting options.

You can configure multiple CAC policies that all apply to a given event (using a Policy Set table type). A nonlimiting option can be given contradictory values in each of these policies. CAC determines what its behavior towards that event is by examining the setting of the option in each applicable policy and applying a rule to produce a “derived value” for the field. If the option is not defined in any policy, then a default behavior is defined. When the SBC is deriving a value for a nonlimiting field, it should disregard all policies in which that field has not been defined by the user. The SBC derives that value based on the assigned behavior for the specific nonlimiting option. The behavior for the nonlimiting options takes one of the following values:

- Last non-default value used. Options of this type take the last non-default value as the derived value. For example, caller-inbound-policy uses the last found non-zero length sdp policy name as the derived value.
- Most restrictive value used. Options of this type take as the derived value the Policy Value that most restricts the behavior of the SBC.
- First non-default value used. Options of this type use the first non-default value as the derived value. For example, caller-voice-qos-profile uses the first non-zero length voice QoS profile name as the derived value.
- All found values combined. Options of this type perform a bitwise-OR to obtain a cumulative value as the derived value.

### Table 7-7  Nonlimiting Options in CAC Entries

<table>
<thead>
<tr>
<th>Nonlimiting Option in a CAC Entry</th>
<th>Behavior of Derived Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>branch bandwidth-field</td>
<td>Last nondefault value used</td>
</tr>
<tr>
<td>branch codec-list</td>
<td>Last nondefault value used</td>
</tr>
<tr>
<td>branch hold-setting</td>
<td>Last nondefault value used</td>
</tr>
<tr>
<td>branch inbound-policy</td>
<td>Last nondefault value used</td>
</tr>
<tr>
<td>branch media-description</td>
<td>All found values combined</td>
</tr>
<tr>
<td>branch media-type</td>
<td>Last nondefault value used</td>
</tr>
</tbody>
</table>
Table 7-7 Nonlimiting Options in CAC Entries (continued)

<table>
<thead>
<tr>
<th>Nonlimiting Option in a CAC Entry</th>
<th>Behavior of Derived Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>branch outbound-policy</td>
<td>Last nondefault value used</td>
</tr>
<tr>
<td>branch privacy</td>
<td>Most restrictive value used</td>
</tr>
<tr>
<td>branch secure-media</td>
<td>All found values combined</td>
</tr>
<tr>
<td>branch sig-qos-profile</td>
<td>First nondefault value used</td>
</tr>
<tr>
<td>branch tel-event payload type</td>
<td>Last nondefault value used</td>
</tr>
<tr>
<td>branch video-qos-profile</td>
<td>First nondefault value used</td>
</tr>
<tr>
<td>branch voice-qos-profile</td>
<td>First nondefault value used</td>
</tr>
<tr>
<td>callee-bandwidth-field</td>
<td>Last nondefault value used</td>
</tr>
<tr>
<td>callee-codec-list</td>
<td>Last nondefault value used</td>
</tr>
<tr>
<td>callee-hold-setting</td>
<td>Last nondefault value used</td>
</tr>
<tr>
<td>callee-inbound-policy</td>
<td>Last nondefault value used</td>
</tr>
<tr>
<td>callee media-description, callee secure media</td>
<td>All found values combined</td>
</tr>
<tr>
<td>callee media-type</td>
<td>Last nondefault value used</td>
</tr>
<tr>
<td>callee-outbound-policy</td>
<td>Last nondefault value used</td>
</tr>
<tr>
<td>callee-privacy</td>
<td>Most restrictive value used</td>
</tr>
<tr>
<td>callee-sig-qos-profile</td>
<td>First nondefault value used</td>
</tr>
<tr>
<td>callee tel-event payload type</td>
<td>Last nondefault value used</td>
</tr>
<tr>
<td>callee video-qos-profile</td>
<td>First nondefault value used</td>
</tr>
<tr>
<td>callee voice-qos-profile</td>
<td>First nondefault value used</td>
</tr>
<tr>
<td>caller-bandwidth-field</td>
<td>Last nondefault value used</td>
</tr>
<tr>
<td>caller-codec-list</td>
<td>Last nondefault value used</td>
</tr>
<tr>
<td>caller-hold-setting</td>
<td>Last nondefault value used</td>
</tr>
<tr>
<td>caller-inbound-policy</td>
<td>Last nondefault value used</td>
</tr>
<tr>
<td>caller media-description, caller secure media</td>
<td>All found values combined</td>
</tr>
<tr>
<td>caller media-type</td>
<td>Last nondefault value used</td>
</tr>
<tr>
<td>caller-outbound-policy</td>
<td>Last nondefault value used</td>
</tr>
<tr>
<td>caller-privacy</td>
<td>Most restrictive value used</td>
</tr>
<tr>
<td>caller sig-qos-profile</td>
<td>First nondefault value used</td>
</tr>
<tr>
<td>caller tel-event payload type</td>
<td>Last nondefault value used</td>
</tr>
<tr>
<td>caller video-qos-profile</td>
<td>First nondefault value used</td>
</tr>
<tr>
<td>codec restrict-to-list</td>
<td>Last nondefault value used</td>
</tr>
<tr>
<td>early media deny</td>
<td>Most restrictive value used</td>
</tr>
<tr>
<td>early media timeout</td>
<td>Most restrictive value used</td>
</tr>
<tr>
<td>early media type</td>
<td>Most restrictive value used</td>
</tr>
</tbody>
</table>
Chapter 7 Implementing Cisco Unified Border Element (SP Edition) Policies

Information About Implementing Policies

Configuring Directed Nonlimiting CAC Policies

In releases prior to Release 3.5.0, you can use the `caller` command and the `callee` command to configure the CAC policy entries that are applied when an adjacency, adjacency group, or account is either a caller or a callee in a call. However, this approach does not permit the configuration of certain directed nonlimiting CAC policy fields on specific adjacencies, adjacency groups, or accounts, in a way that is independent of whether the adjacencies, adjacency groups or accounts are the callees or the callers on the calls. The following example illustrates this limitation.

Suppose the following sequence of commands is part of the configuration of an entry in a CAC table:

```
cac-policy-set 3
  first-cac-table cac-tb1
  cac-table cac-tb1
  table-type limit adjacency
  entry 1
  match-value adj1
caller port-range-tag adj-name
  callee port-range-tag adj-name
  action cac-complete
```

If there is a call from the adj1 adjacency to the adj2 adjacency, the settings specified for the caller in this example is applied to adj1. At the same time, the callee settings are applied to adj2 because that adjacency is the callee in this call. In a scenario such as this one, you might not want to apply any configuration to the other adjacency (the adj2 adjacency, in this example) involved in the call. The `branch` command helps overcome this limitation. This command has been introduced in Release 3.5.0.

In the preceding example, the `branch` command can be used to replace the `caller` command and the `callee` command as follows:

```
cac-policy-set 3
  first-cac-table cac-tb1
  cac-table cac-tb1
  table-type limit adjacency
  entry 1
  match-value adj1
  branch port-range-tag adj-name
  action cac-complete
```

The `branch` command is not a replacement for the `caller` command and the `callee` command pair in scenarios in which you want to apply settings to both the caller adjacency and the callee adjacency.

With this configuration, the settings specified in the `branch` command are applied to the adj1 adjacency. For a call from the adj2 adjacency to the adj1 adjacency, the same settings are applied to the adj1 adjacency. For this call, no settings are applied to adj2 or any other adjacency that calls or is called by adj1.

---

### Table 7-7 Nonlimiting Options in CAC Entries (continued)

<table>
<thead>
<tr>
<th>Nonlimiting Option in a CAC Entry</th>
<th>Behavior of Derived Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>media address preserve, media bandwidth-field ignore, media tel-event interworking</td>
<td>All found values combined</td>
</tr>
<tr>
<td>sdpm-media-profile</td>
<td>Last nondefault value used</td>
</tr>
<tr>
<td>transcode-deny</td>
<td>Most restrictive value used</td>
</tr>
<tr>
<td>transport srtpt</td>
<td>Most restrictive value used</td>
</tr>
</tbody>
</table>

**Note:**

The `branch` command is not a replacement for the `caller` command and the `callee` command pair in scenarios in which you want to apply settings to both the caller adjacency and the callee adjacency.
CHAPTER 7
Implementing Cisco Unified Border Element (SP Edition) Policies

Information About Implementing Policies

The following are the features of the **branch** command:

- If a branch setting (that is, the **branch** command) and a caller-callee pair setting (that is, the **caller** and **callee** command pair) are configured in different policy entries, the setting in the last entry of the configuration takes precedence.
- If two branch settings, each in a different policy entry, are encountered, the setting in the last entry that is encountered takes precedence.
- If a branch setting and a caller-callee setting are in the same policy entry, the branch setting takes precedence over the caller-callee setting.

The following sample configuration illustrates how the branch command works:

```plaintext
cac-policy-set 3
  first-cac-table cac-tbl
  cac-table cac-tbl
  table-type limit adjacency
  entry 1
    match-value phone2
    branch port-range-tag adj-name
    caller port-range-tag string tagB_cac
    callee port-range-tag string tagA_cac
    action cac-complete
  entry 2
    match-value phone1
    branch port-range-tag string tagA_cac
    caller port-range-tag adj-name
    callee port-range-tag adj-name
    action cac-complete
  complete
  cac-policy-set global 3
media-address ipv4 209.165.202.130
  port-range 10000 15000 any
  port-range 15002 15003 any tag phone1
  port-range 16002 16003 any tag phone2
  port-range 17002 17003 any tag tagA_cac
  port-range 18002 18003 any tag tagB_cac
```

In this example, the call goes from phone 1 to phone 2. The following sequence of events takes place during the call:

1. Matching is performed on the source adjacency, phone 1, which matches entry 2. Here, the branch entry refers to the caller side, so the caller entry is overridden. After this first policy match is performed, port-range-tag is set to tagA_cac on side A. In addition, the callee port tag is set to adj-name.

2. Matching is performed on the destination adjacency, phone2, which matches entry 1. Here, the branch entry refers to the callee side, so the callee entry is overridden. This entry sets the caller side port-range-tag to tagB_cac. In other words, adj_name is assigned as the callee side port-range-tag. These settings take precedence over the values assigned in the previously matched entry, entry 2, because these settings are assigned later.

The outcome is that the tagB_CAC port is used on side A, and an adj-name port, phone2, is used on side B.

**Media Line Removal**

Media line removal feature provides the ability to strip or pad disabled media descriptions (m-lines with zero port) when sending an offer or answer to interoperate with various non-compliant devices.
Where the SDP being forwarded represents an answer, the media line which was removed from the forwarded offer is identified and a dummy media line is inserted into the same location. This is required for the compliant partner to match appropriate media line requests and responses.

Where the SDP being forwarded is a future offer, it uses offer modification to effectively shuffle-up media lines allowing the “padding” dummy media lines to be added to the end of the forwarded SDP.

SBC’s transmit behavior is independently configured for the caller and callee sides of the call using the following options:

- strip new on offer—removes disabled media streams in forwarded offers which are new or unknown to the recipient of the offer.
- strip all on offer—removes all disabled media streams from forwarded offers, whether known to the recipient of the offer or not.
- strip on answer—removes all disabled media streams from forwarded answers.
- do not pad on offer—stops SBC from padding forwarded offers with disabled media streams. This means that a forwarded offer may not comply because it may contain less media lines than previous offers.

**Note**

The “strip new on offer” and “strip all on offer” result in removal of m-lines from the forwarded offer. The missing lines are not “padded in” and there is no need to set the “do not pad on offer” option to achieve this. The “do not pad on offer” option only affects media lines that were missing from the received offer.

On selecting the appropriate option, the SDP to be forwarded is created with disabled media portions deleted, rather than the existing behavior of setting the port to zero.

**Multiple SBC Media Bypass**

The multiple SBC media bypass feature can send media packets directly from the answerer to the original offerer. When the SBC detects that the media packets are being looped back unnecessarily, as shown in Figure 7-5, the SBC removes itself from the loop so that the media packets can flow directly between the endpoints.
Partial Media Bypass

When at least one SBC from the network has to anchor the media because endpoints cannot communicate directly, the other SBC gets bypassed as shown in Figure 7-6. If the media bypass type is explicitly configured to be partial, only IP realm and VPN configuration on the adjacency can be used to determine whether media bypass is possible. Because media bypass tags are not used, the VPN names must be globally unique across all the SBCs for partial media bypass to work.

**Figure 7-6 Partial Media Bypass**

![Partial Media Bypass Diagram](image-url)
Figure 7-7 shows an example of media bypass across two or more SBC devices.

In networks where direct media packets cannot pass, the feature creates an optimized media path through a group of SBCs, to avoid unnecessary media hops through the SBC network.

With the multiple SBC media bypass feature, the SBC can transmit an extra set of media addresses alongside an SDP offer. These are the original media addresses that the SBC itself received from the offerer. The original media addresses are placed in a separate multiple SBC media bypass feature information element. These addresses are associated with information about the media plane connectivity of the offerer. A downstream SBC uses the multiple SBC media bypass feature connectivity information to determine whether it can re-instate the original media addresses by rewriting the SDP offer to include them. This enables the media packets to directly pass between the answerer and the original offerer.

The multiple SBC media bypass feature information can also be used by a group of SBCs to optimize the media path and to avoid unnecessary media hops through the SBC network. The SDP answer is accompanied by an indication of whether the feature was successful or not. The SBC uses this indication to determine whether it has been bypassed or whether it is still in the media path. When many SBCs appear in the media path, they collectively build up a stack of alternative media addresses for each media streams in the offer, where each element of the stack has associated connectivity information.

The SBCs determines which endpoints and intermediate hops are connected to decide which intermediate entities can be bypassed. Such connectivity information is passed on by tags in the multiple SBC media bypass feature information elements. Two remote endpoints on an adjacency can be connected if they have one or more matching tags. Therefore, tags must be globally unique for the multiple SBC media bypass feature protocol to work.
Continuing Media Bypass After a Session Refresh

When a media bypass call is in progress, the SIP registrar does not process the media exchanged by the endpoints. Therefore, the registrar uses signaling to detect failures in the session. The registrar sends a session refresh request to check whether a session is alive. The session refresh request is in the form of an INVITE or UPDATE message containing a copy of the SDP forwarded by the registrar during the original call setup.

When the SBC receives the INVITE message from the SIP registrar, it does not correlate the SDP in the message with the SDP sent earlier in the call. The SBC processes the SDP in the INVITE message as normal and creates an SDP offer to send to either the caller endpoint or callee endpoint. From the perspective of the endpoint, the INVITE message is an attempt to renegotiate the media for the call. The endpoint processes the offer and creates an answer that is consistent with the offer. This answer is returned to the registrar through the SBC.

In the answer, the port number for each media stream in the call is different from the port number of the previous media stream. This mismatch in the port number could cause the registrar to send a late INVITE message to the endpoint. An endpoint that does not support the receipt of a late INVITE message for a renegotiation would reject the message. The call fails because the media is being sent from the endpoint to the SBC, from where the media is dropped. To circumvent this issue, renegotiation is enabled by default so that the same path is used to resume exchange of media packets between the endpoints. This feature ensures that media bypass calls continue to bypass the media after a session refresh.

Note

You can disable or enable renegotiation.

Restrictions

The multiple SBC media bypass feature has the following restrictions:

- Media bypass is not supported for H.323 calls.
- Media services, such as provisioned transcoding, transrating, and DTMF Interworking preclude media, are sent directly between endpoints. For a given call, if the administrator has configured media bypass settings and if media bypass is possible, then it takes precedence over other media services. However, if lawful intercept (LI) is provisioned on the SBC, LI would take precedence over the multiple SBC media bypass feature.
- The SBC does not support the feature when one endpoint is IPv4 and the other endpoint is IPv6. Because the endpoints cannot understand the traffic they receive.
- The SBC does not support the feature when one endpoint is SIP and the other endpoint is H.323 if SIP-H.323 interworking is enabled. Because the endpoints cannot understand the traffic they receive.

Performance Impact

When the multiple SBC media bypass feature is enabled, it has the following performance impact on the Cisco ASR 1000 series routers:

- The SBCs signaling performance decreases by a small fraction, due to the increased parsing and message manipulation costs. However there is a corresponding gain on media resources for every call that successfully negotiates the feature.
- The transient occupancy of each call setup increases by a multiple of the size of the multiple SBC media bypass feature information that is encoded in the SIP message, plus a small amount of control information. For SDP sizes of 300 bytes, this is predicted to be around 1500 bytes in total. However,
the steady-state occupancy for calls that successfully negotiate the multiple SBC media bypass feature decreases as no media resources are required for those calls. This saves approximately 10000 bytes per call.

For more information on configuring the multiple SBC media bypass feature, see the ?$paranum>Configuring Multiple SBC Media Bypass? section on page 7-136. For configuration examples of the feature, see the ?$paranum>Example: Multiple SBC Media Bypass? section on page 7-161. For video of the example that explains how the SBC Media Bypass feature works, see http://www.cisco.com/en/US/docs/routers/asr1000/configuration/guide/SBCU3.5S/sbc_media_bypass.html.

Common IP Address Media Bypass

This section contains the following topics:

- Restrictions for Common IP Address Media Bypass, page 7-43
- Information About Common IP Address Media Bypass, page 7-43
- Features of Common IP Address Media Bypass, page 7-44

Restrictions for Common IP Address Media Bypass

The following are restrictions for the Common IP Address Media Bypass feature:

- This feature is not supported for H.323 adjacencies.
- This feature is not supported in a scenario in which endpoints are behind the same NAT device but are not registered with the SBC.
- This feature is not supported in a scenario in which the caller endpoint and callee endpoint are behind different NAT devices even when there is connectivity between the networks defined by each NAT device. The SBC always relays media between two such endpoints.
- This feature is not supported in a scenario in which only one of the endpoints is behind a NAT device. The SBC always relays media between two such endpoints.

Information About Common IP Address Media Bypass

When you enable the Multiple SBC Media Bypass feature, the SBC bypasses or relays media between the endpoints of an adjacency depending on the media bypass tags presented by the endpoints. If the tags match, the SBC determines that there is media connectivity between the endpoints and, therefore, bypasses itself from the media flow between the endpoints. In contrast, if the tags do not match, the SBC determines that there is no media connectivity between the endpoints and, therefore, relays media between the endpoints.

Note

For detailed information about the Multiple SBC Media Bypass feature, see the ?$paranum>Multiple SBC Media Bypass? section on page 7-39.

An organization can use a hosted PBX solution that is owned and managed by a service provider. Typically, a hosted PBX solution can serve many organizations and, therefore, serve multiple NAT devices. There may be a scenario in which there are multiple NAT devices behind a single adjacency. In such a scenario, the SBC must bypass media for the endpoints behind the same NAT device and relay media for the endpoints that are behind different NAT devices. In releases prior to Release 3.6.0, the only way to achieve this is to configure an adjacency for each NAT device. This approach increases the overhead involved in managing the network.
The Common IP Address Media Bypass feature is an enhancement to the Multiple SBC Media Bypass feature. It offers an alternative to the approach of creating an adjacency for each NAT device.

When you configure the Common IP Address Media Bypass feature, the SBC assigns each endpoint behind a NAT device a media bypass tag that is based on the corresponding endpoint’s external, NAT IP address. These media bypass tags are used by the SBC to determine whether the caller endpoint and callee endpoint belong to the same NAT network. If the media bypass tag of the caller endpoint matches the media bypass tag of the callee endpoint, the SBC bypasses media. If the tags do not match, the SBC relays media.

**Note**
The Common IP Address Media Bypass feature does not introduce any change in the mechanism by which the SBC compares the media bypass tags of the caller endpoint and callee endpoint. In other words, the SBC does not distinguish between the feature or method by which media bypass tags are created. The SBC only compares the tags and bypasses media when the tags match.

When the Common IP Address Media Bypass feature is not configured or is disabled, media-bypass decisions are taken by the SBC on the basis of the media bypass tags configured at the adjacency level. If media bypass tags are not configured, media-bypass decisions are taken on the basis of the autogenerated tags that are based on VPN IDs.

When the Common IP Address Media Bypass feature is configured and enabled:

- If the caller endpoint or callee endpoint is a registered subscriber that has been identified at registration time as being behind a NAT device, the SBC generates a media bypass tag and uses that media bypass tag for the call leg.
- If the caller endpoint or callee endpoint is not a registered subscriber or is not behind a NAT device, the SBC uses the tag that is created by the `media bypass tag` command, if such a tag is present, for the call leg.
- If the caller endpoint or callee endpoint is not a registered subscriber or is not behind a NAT device and if there are no configured tags, the SBC generates a media bypass tag based on the VPN ID of the endpoint, for the call leg.

During the call, if both the endpoints have media bypass tags that match, the SBC determines that both the endpoints are behind the same NAT device and it bypasses media for that call leg. Conversely, if the media bypass tags do not match, if either endpoint does not have a media bypass tag, or if either endpoint is not a registered subscriber, the SBC relays media for that call leg.

The following is the format of the media bypass tag generated by this feature:

```
nat-VPN-ID-IP-address
```

In this format, `IP-address` is the source IP address of the most recent non-fast-patched, successful REGISTER request from the endpoint. The IP address can be in IPv4 format or IPv6 format.

The following are sample media bypass tags generated by this feature:

- nat-123-192.0.2.6
- nat-254-192.0.26.18
- nat-2233-2001:DB8::AC10:FE01

### Features of Common IP Address Media Bypass

The following are additional points about how the Common IP Address Media Bypass feature works:

- After this feature is configured, the SBC can detect whether an endpoint is behind a NAT device by using the existing adjacency configuration features:
If the no nat command is configured for an adjacency, an endpoint behind that adjacency is recognized as being behind a NAT if the IP address in the Via header of the SIP messages from that adjacency is different from the IP address from which the request was received.

- If the nat force-on command is configured, all endpoints are assumed to be behind a NAT.

- If the nat force-on command is configured and an endpoint is not behind a NAT, the SBC relays media for calls to and from such an endpoint. Note that if this feature is disabled, the SBC bypasses media. When you enable this feature, the SBC starts relaying media.

- If this feature is configured while an adjacency is active, only new calls that are processed by that adjacency are affected by this feature. Existing calls are not affected.

- This feature is independent of whether the initial INVITE message contains SDP content because the media bypass tag is not added to the SDP content.

- This feature is supported by all forms of media bypass:
  - Simple media bypass, in which the caller endpoint and callee endpoint are associated with local adjacencies.
  - Two-call media bypass, in which the SBC forwards the call to a softswitch or registrar, which then loops the call request back to the SBC.
  - N-call media bypass, in which a call is looped through the SBC multiple times.
  - Multi-SBC media bypass, in which a call is looped through multiple SBCs.

### CAC Rate Limiting

You can limit the number or the rate of new calls accepted and the number of media renegotiations within a call. However, limits are not placed on the following:

- Media renegotiations which do not actually change the characteristics of the call.
- Any other in-call messages.

In-call messages include any message within the context of a call, including provisional responses during call setup and call renegotiation messages, but not including call setup or tear-down messages.

- Internally-generated messages

**Note** You cannot specify limits at the granularity of a specific SIP or H.323 message.

You can also limit the rate and number of registrations passing through the Cisco Unified Border Element (SP Edition). However, limits are not placed on any other out-of-call messages. (An out-of-call message is any messages which is not following within the context of a call and which does not form part of registration processing. These are always classified as either a request or a response.)

You can rate limit all in-call and out-of-call messages.

This includes in-call messages at all scopes, as normal. For example:

- Configuration at the “per-call” scope allows you to limit the rate at which an endpoint sends messages within a call.
- Configuration at the “dst-adjacency” scope allows you to limit the total rate of in-call messages sent out of an adjacency within all of the calls using that adjacency. (This could ensure that the load out of an adjacency never exceeds that which the attached network entity can cope with.)

The following messages are not rate-limited:
- SIP INVITE requests: 200 responses and ACK messages
- SIP PRACK messages and response
- SIP BYE messages and responses
- Any SIP message with non-duplicate SDP on
- For H.323 calls: Q.931 SETUP, Q.931 CONNECT and Q.931 RELEASE messages.

You can place restrictions on the rate at which out-of-call messages are processed. Configuration is permitted at all scopes except per-call scope (because this scope does not exist for out-of-call messages).

The Cisco Unified Border Element (SP Edition) will gracefully reject in-call messages when the rate exceeds that specified in the CAC. When an in-call message is not processed, the Cisco Unified Border Element (SP Edition) does the following:

- For SIP messages, Cisco Unified Border Element (SP Edition) rejects the message gracefully wherever possible. The rejection is sent back to the sending endpoint, so the call is likely to survive.
- For H.323 messages, Cisco Unified Border Element (SP Edition) drops the message because they usually cannot be gracefully rejected. This is likely to be disruptive for the call.

The Cisco Unified Border Element (SP Edition) gracefully rejects out-of-call messages when the rate exceeds that specified in CAC.

All rate limits must be protocol stack independent; limits must police SIP and H.323 messages.

In addition to configuring blacklists based on a number of CAC policy failures, you can now allow blacklists to be applied to endpoints that send in-call or out-of-call messages at a high rate.

### Multiple CAC Averaging Periods

The user can apply different rate limits over a different averaging period by configuring a second set of rate-limiting CAC criteria. The user is able to do the following:

- Set the averaging period for the secondary rate calculation.
- Set the maximum number of new calls per minute for the secondary rate calculation, if a limit is required.
- Set the maximum number of endpoint registrations per minute for the secondary rate calculation, if a limit is required.
- Set the maximum number of in-call messages to be processed per minute for the secondary rate calculation, if a limit is required.
- Set the maximum number of out-of-call messages to be processed per minute for the secondary rate calculation, if a limit is required.

The user can configure two sets of SBC policies together that have rate-limiting criteria. The CAC rejects an event if it breaks any of the configured limits.

### Subscriber Policy

A user can subscribe multiple endpoints to the network to allow them to make calls. A subscriber is one of those endpoints. In a particular network, you might want to limit each subscriber to no more than a specific number of simultaneous calls. The Subscriber Policy feature allows you to limit each subscriber to a specific number of simultaneous calls.
This feature provides the ability to configure the CAC limits. For example, you can configure the maximum number of concurrent calls, the maximum number of registrations, or the maximum call rate at different scopes, such as subscriber, subscriber category, and subscriber category prefix.

You can configure CAC tables:

- To associate a subscriber with a subscriber category. Call events between that subscriber and the core network are also associated with that same subscriber category.
- To match on a subscriber category or on a subscriber category prefix (the first n bits of the subscriber category), and then set limits when matched. The subscriber category prefix specifies the length of prefix to match. If specified, then only the first n bits of each of the call's subscriber categories is checked for a match.
- To set limits per subscriber category.
- To set limits per subscriber.

Note that when a subscriber scope is enabled, the SBC tracks an additional group of ALL "non-subscribers." The non-subscriber group is matched if the call is from a non-subscriber. Limits set in the subscriber scope apply to this non-subscriber group.

Privacy Service

The SBC provides the privacy service to ensure that requests for anonymity, as requested by a user during signaling, can be dynamically acted upon to ensure that the user’s anonymity is maintained when the user leaves a trusted network. A user can request various levels of anonymity, with the privacy service removing the information that a user wants to withhold. The SBC can be configured such that individual adjacencies can be marked as trusted, untrusted, or configured in order to apply the privacy service. The privacy service is applied in a CAC policy set.

In addition to this, the SBC can edit—override or modify—a user’s request for privacy when forwarding the privacy request. For example, a user can request identity of self to be withheld, but by editing the privacy request, the identity can be provided.

A user can also provide indications of anonymity in the display and presentation number. During number analysis, these calls can be detected and different analysis trees be used to progress the call.

The Privacy Service feature provides the following functions:

- Apply a privacy service based on information provided by a user when leaving a trusted domain.
- Edit a privacy service on request from a user and perform functions such as pass, strip, insert, and replace indications.
- Declare configurable trust boundaries.
- Detect calls in number analysis where the source is anonymous.
- Standard SIP header rewriting is performed by the SBC to cover the additional requirements specified in the SIP privacy header:
  - The Call ID, Server, and Contact headers are rewritten to hide the endpoint's identity.
  - Any Via headers are cached and replaced on the message with a single header identifying the SBC.

Both SIP and H323 adjacencies allow the configuration of the trusted and untrusted statuses.

For information about configuring the Privacy Service feature, see the Configuring Privacy Service? section on page 7-126.
Session Initiation Protocol

In the context of SIP, a user indicates the levels of privacy that should be applied using the Privacy header. If the SBC cannot recognize any of the tokens present in the header, the message is rejected with a 433 Anonymity Disallowed response. Similarly, a response containing a critical privacy request that cannot be met is converted to a 433 failure response for an in-call message. For an out-of-dialog message, the response is dropped to ensure that no private information gets leaked accidently.

If this is an in-call message that does not contain a privacy header, the privacy requirements are assumed to be the same as those specified in the last privacy header from the side of the call. However, if the SBC reroutes a call locally, for example, a SIP 3xx redirect response, it discards the previously learnt privacy requirements on the side of the call that has been rerouted.

The following events occur when privacy services are applied to a request or response:

- When the privacy service based on a user, Privacy: user, is applied to a request or response, the Reply-To, Call-Info, User-Agent, Organization, Subject, In-Reply-To, Warning, and Server headers are stripped from the message.
  
  Also, when the privacy service based on a user, Privacy: user, is applied to a request, the URI in the From header is rewritten to anonymous@anonymous.invalid. The original URI is stored for replacement on responses. The display name in the From header is removed, and any further header manipulation rules that are configured as part of the user ID privacy are applied to the message.

- When the privacy service based on ID value, Privacy: id, is applied to a request or a response, the P-Preferred-ID, P-Asserted-Identity, and Remote-Party-Id headers are stripped from the message.

- When the privacy service based on session privacy, Privacy: session, is applied to a request or a response, media bypass is disallowed. However, if the session privacy is critical, and it is too late to disable media bypass, the call is torn down.

- When the privacy service based on header privacy, Privacy:header, is applied to a request or response, Record-Route or Route headers, if any, are removed and stored. They are restored on the responses within the dialog. If any further header manipulation rules are configured, they are applied to the message. The SBC strips the Privacy header from the ongoing message and removes the privacy option-tag, if any, from the Proxy-Require header.

Users can dynamically request for privacy service. This service can be applied by inserting Privacy: header based on RFC 3323 and RFC 3325.

Privacy Service on SIP Requests

Table 7-8 lists the behavior of the privacy service when it is applied on SIP requests, and Privacy: header is present to indicate the appropriate level of privacy to be applied.

<table>
<thead>
<tr>
<th>Header Name</th>
<th>None</th>
<th>User</th>
<th>Header</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
<td>—</td>
<td>Set to anonymous value: <a href="mailto:anonymous@anonymous.invalid">anonymous@anonymous.invalid</a></td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>Contact</td>
<td>—</td>
<td>—</td>
<td>Rewritten</td>
<td>—</td>
</tr>
<tr>
<td>Reply-to</td>
<td>—</td>
<td>Stripped</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>Via</td>
<td>Stripped</td>
<td>Stripped</td>
<td>Stripped</td>
<td></td>
</tr>
<tr>
<td>Call-Info</td>
<td>—</td>
<td>Stripped</td>
<td>—</td>
<td></td>
</tr>
</tbody>
</table>
Privacy Service on SIP Responses

Table 7-9 lists the behavior of the privacy service when it is applied on SIP responses.

<table>
<thead>
<tr>
<th>Header Name</th>
<th>None</th>
<th>User</th>
<th>Header</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Contact</td>
<td>—</td>
<td>—</td>
<td>Stripped</td>
<td>—</td>
</tr>
<tr>
<td>Reply-to</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Via</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Call-Info</td>
<td>—</td>
<td>Stripped</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>User-Agent</td>
<td>—</td>
<td>Stripped</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Organization</td>
<td>—</td>
<td>Stripped</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Server</td>
<td>—</td>
<td>Stripped</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Subject</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Call-ID</td>
<td>Rewritten</td>
<td>Rewritten</td>
<td>Rewritten</td>
<td>Rewritten</td>
</tr>
<tr>
<td>In-Reply-To</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Warning</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>P-Asserted-Identity</td>
<td>—</td>
<td>—</td>
<td>Stripped</td>
<td>—</td>
</tr>
<tr>
<td>P-Preferred-Identity</td>
<td>—</td>
<td>—</td>
<td>Stripped</td>
<td>—</td>
</tr>
<tr>
<td>Remote-Party-ID</td>
<td>—</td>
<td>—</td>
<td>Stripped</td>
<td>—</td>
</tr>
<tr>
<td>Record-Route</td>
<td>—</td>
<td>—</td>
<td>Stripped</td>
<td>—</td>
</tr>
</tbody>
</table>
Privacy Service on H.323

The SBC treats the following H.323 protocol events as requests for the privacy service:

- On a Q.931 Setup, the caller address presentation restriction is requested if the Q.931 callingPartyNumber is present, and contains a presentationIndicator set to 3, presentation restricted, or, the H.225 presentationIndicator is present and set to presentationRestricted.

- On a Q.931 Connect, callee address presentation restriction is requested if the Q.931 connectedNumber is present, and contains a presentationIndicator set to 3, presentation restricted, or, the H.225 presentationIndicator is present and set to presentationRestricted.

When there is a conflict between the two presentationIndicators, the value in the Q.931 callingPartyNumber, the connectedNumber, takes precedence.

H.323 to SIP

A presentation restriction indication that is received for the callingPartyNumber or connectedNumber elements in an H.323 message is considered a request for header;id;critical privacy when being translated to a SIP privacy request.

If the presentation restriction is requested by the H.323 side, the URI in the From header is rewritten with anonymous@anonymous.invalid whether or not the SBC is acting as a privacy service.

When interworking with the SIP, the privacy service is always applied.

SIP to H.323

A SIP-signaled request for id or header privacy is translated into an H.323 presentation restriction on outgoing addresses, if any. All the other SIP privacy tokens are ignored.

Message, Policy, and Subscriber Statistics

From Cisco IOS XE Release 3.3S, enhancements have been made to the following statistics:

- Call Statistics, page 7-50
- CAC Statistics, page 7-55
- Subscriber Statistics, page 7-58

Call Statistics

The call-related statistics have been enhanced to include the following two features:

- Number of calls completed during a period—The summary periods of the call-related statistics includes the total number of calls that have been completed. A call is completed because of a signaling message received from an upstream or a downstream device. For SIP calls, a call is completed when a participating endpoint sends a BYE message. For H.323 calls, a call is completed when a participating endpoint sends a ReleaseComplete message with causeValue of 16, which indicates a normal call clearing.

- Running total for the calls statistics—The summary periods of the call-related statistics includes a current-indefinite time value. This time value provides the call-statistics for a period since the value has been last reset. Initially, the current-indefinite time value displays the statistics for the time
The following commands are used for displaying the call-related statistics and resetting the call-related
statistics:

- The `show sbc sbc-name sbe call-stats {all | global | per-adjacency adjacency-name | src-account
  name | dst-account name | src-adjacency name | dst-adjacency name} period` command—Lists the
  statistics pertaining to all the calls on a SBE for a particular period, such as current-indefinite.

- The `clear sbc sbc-name sbe call-stats {all | dst-account account-name | dst-adjacency
  adjacency-name | global | src-account account-name | src-adjacency adjacency-name | per-adjacency
  adjacency-name} [current-indefinite]` command—Clears the call statistics on a SBE by the current-indefinite period.

The following example shows how the `show sbc sbc-name sbe call-stats all current-indefinite` command
displays statistic pertaining to all calls on the SBE for the current-indefinite period:

```plaintext
Router# show sbc SBC2 sbe call-stats all current-indefinite

statistics for the current indefinite for source adjacency phone1

Call count totals:
  Total call attempts = 1
  Total active calls = 0
  Total active IPv6 calls = 0
  Total activating calls = 0
  Total de-activating calls = 0
  Total active emergency calls = 0
  Total active e2 emergency calls = 0
  Total IMS rx active calls = 0
  Total IMS rx call renegotiation attempts = 0
  Total SRTP-RTP interworked calls = 0
  Total active calls not using SRTP = 0
  Total active transcoded calls = 0
  Total active transrated calls = 0
  Total calls completed = 1

General call failure counters:
  Total call setup failures = 0
  Total active call failures = 0
  Total failed call attempts = 0
  Total failed calls due to update failure = 0
  Total failed calls due to resource failure = 0
  Total failed calls due to congestion = 0
  Total failed calls due to media failure = 0
  Total failed calls due to signaling failure = 0
  Total failed calls due to IMS rx setup failure = 0
  Total failed calls due to IMS rx renegotiation failure = 0
  Total failed calls due to RTSP disallowed on call leg = 0
  Total failed calls due to SRTP disallowed on call leg = 0

Policy control failures:
  Call setups failed due to NA = 0
  Call setups failed due to RTG = 0
  Call setups failed due to CAC = 0
  CAC fails due to number of calls limit = 0
  CAC fails due to call rate limit = 0
  CAC fails due to bandwidth limit = 0
  CAC fails due to number of media channels limit = 0
  CAC fails due to number of media update limit = 0
  CAC message drops due to mid call message rate limit = 0
  CAC message drops due to out of call message rate limit = 0
```
Stats Reset Timestamp:
Timestamp when stats for this summary period were reset = 2011/03/07 03:27:36
statistics for the current indefinite for destination adjacency phone2

Call count totals:
Total call attempts = 1
Total active calls = 0
Total active IPv6 calls = 0
Total activating calls = 0
Total de-activating calls = 0
Total active emergency calls = 0
Total active e2 emergency calls = 0
Total IMS rx active calls = 0
Total IMS rx call renegotiation attempts = 0
Total SRTP-RTP interworked calls = 0
Total active calls not using SRTP = 0
Total active transcoded calls = 0
Total active transrated calls = 0
Total calls completed = 1

General call failure counters:
Total call setup failures = 0
Total active call failures = 0
Total failed call attempts = 0
Total failed calls due to update failure = 0
Total failed calls due to resource failure = 0
Total failed calls due to congestion = 0
Total failed calls due to media failure = 0
Total failed calls due to signaling failure = 0
Total failed calls due to IMS rx setup failure = 0
Total failed calls due to IMS rx renegotiation failure = 0
Total failed calls due to RTP disallowed on call leg = 0
Total failed calls due to SRTP disallowed on call leg = 0

Policy control failures:
Call setups failed due to NA = 0
Call setups failed due to RTG = 0
Call setups failed due to CAC = 0
CAC fails due to number of calls limit = 0
CAC fails due to call rate limit = 0
CAC fails due to bandwidth limit = 0
CAC fails due to number of media channels limit = 0
CAC fails due to number of media update limit = 0
CAC message drops due to mid call message rate limit = 0
CAC message drops due to out of call message rate limit = 0

Stats Reset Timestamp:
Timestamp when stats for this summary period were reset = 2011/03/07 03:27:36
statistics for the current indefinite for source account source1

Call count totals:
Total call attempts = 1
Total active calls = 0
Total active IPv6 calls = 0
Total activating calls = 0
Total de-activating calls = 0
Total active emergency calls = 0
Total active e2 emergency calls = 0
Total IMS rx active calls = 0
Total IMS rx call renegotiation attempts = 0
Total SRTP-RTP interworked calls = 0
Total active calls not using SRTP = 0
Total active transcoded calls = 0
Total active transrated calls = 0
Total calls completed = 1
General call failure counters:
Total call setup failures = 0
Total active call failures = 0
Total failed call attempts = 0
Total failed calls due to update failure = 0
Total failed calls due to resource failure = 0
Total failed calls due to congestion = 0
Total failed calls due to media failure = 0
Total failed calls due to signaling failure = 0
Total failed calls due to IMS rx setup failure = 0
Total failed calls due to IMS rx renegotiation failure = 0
Total failed calls due to RTP disallowed on call leg = 0
Total failed calls due to SRTP disallowed on call leg = 0

Policy control failures:
Call setups failed due to NA = 0
Call setups failed due to RTG = 0
Call setups failed due to CAC = 0
CAC fails due to number of calls limit = 0
CAC fails due to call rate limit = 0
CAC fails due to bandwidth limit = 0
CAC fails due to number of media channels limit = 0
CAC fails due to number of media update limit = 0
CAC message drops due to mid call message rate limit = 0
CAC message drops due to out of call message rate limit = 0

Stats Reset Timestamp:
Timestamp when stats for this summary period were reset = 2011/03/07 03:27:36
Statistics for the current indefinite for destination account dest1
Call count totals:
Total call attempts = 1
Total active calls = 0
Total active IPv6 calls = 0
Total activating calls = 0
Total de-activating calls = 0
Total active emergency calls = 0
Total active e2 emergency calls = 0
Total IMS rx active calls = 0
Total IMS rx call renegotiation attempts = 0
Total SRTP-RTP interworked calls = 0
Total active calls not using SRTP = 0
Total active transcoded calls = 0
Total active transrated calls = 0
Total calls completed = 1

General call failure counters:
Total call setup failures = 0
Total active call failures = 0
Total failed call attempts = 0
Total failed calls due to update failure = 0
Total failed calls due to resource failure = 0
Total failed calls due to congestion = 0
Total failed calls due to media failure = 0
Total failed calls due to signaling failure = 0
Total failed calls due to IMS rx setup failure = 0
Total failed calls due to IMS rx renegotiation failure = 0
Total failed calls due to RTP disallowed on call leg = 0
Total failed calls due to SRTP disallowed on call leg = 0

Policy control failures:
Call setups failed due to NA = 0
Call setups failed due to RTG = 0
Call setups failed due to CAC = 0
Message, Policy, and Subscriber Statistics

CAC fails due to number of calls limit = 0
CAC fails due to call rate limit = 0
CAC fails due to bandwidth limit = 0
CAC fails due to number of media channels limit = 0
CAC fails due to number of media update limit = 0
CAC message drops due to mid call message rate limit = 0
CAC message drops due to out of call message rate limit = 0

Stats Reset Timestamp:
Timestamp when stats for this summary period were reset = 2011/03/07 03:27:36

statistics for the current indefinite for global counters

Call count totals:
Total call attempts = 1
Total active calls = 0
Total active IPv6 calls = 0
Total activating calls = 0
Total de-activating calls = 0
Total active emergency calls = 0
Total active e2 emergency calls = 0
Total IMS rx active calls = 0
Total IMS rx call renegotiation attempts = 0
Total SRTP-RTP interworked calls = 0
Total active calls not using SRTP = 0
Total active transcoded calls = 0
Total active transrated calls = 0
Total calls completed = 1

General call failure counters:
Total call setup failures = 0
Total active call failures = 0
Total failed call attempts = 0
Total failed calls due to update failure = 0
Total failed calls due to resource failure = 0
Total failed calls due to congestion = 0
Total failed calls due to media failure = 0
Total failed calls due to signaling failure = 0
Total failed calls due to IMS rx setup failure = 0
Total failed calls due to IMS rx renegotiation failure = 0
Total failed calls due to RTP disallowed on call leg = 0
Total failed calls due to SRTP disallowed on call leg = 0

Policy control failures:
Call setups failed due to NA = 0
Call setups failed due to RTG = 0
Call setups failed due to CAC = 0
CAC fails due to number of calls limit = 0
CAC fails due to call rate limit = 0
CAC fails due to bandwidth limit = 0
CAC fails due to number of media channels limit = 0
CAC fails due to number of media update limit = 0
CAC message drops due to mid call message rate limit = 0
CAC message drops due to out of call message rate limit = 0

Stats Reset Timestamp:
Timestamp when stats for this summary period were reset = 2011/03/07 03:27:36

statistics for the current indefinite for adjacency phone1

Stats Reset Timestamp:
Timestamp when stats for this summary period were reset = 2011/03/07 03:27:36

Current count of Media Packets Lost = 0
Current count of Media Packets Dropped = 0
Current count of Media Packets Sent = 236
Current count of Media Packets Received = 236
Current count of RTCP Packets Sent = 0
Current count of RTCP Packets Received = 0
Average Call Duration = 22004
Average of the Answer Seizure Ratio = 0
Average of the Round Trip Delay = 0 ms
Average of the locally calculated jitter = 0 ms
Average of the remotely calculated jitter = 0 ms
Average of the received media dropped per thousand pkts = 0
Average of the sent media lost per thousand pkts = 0

Statistics for the current indefinite for adjacency phone2

Stats Reset Timestamp:
Timestamp when stats for this summary period were reset = 2011/03/07 03:27:36
Current count of Media Packets Lost = 0
Current count of Media Packets Dropped = 0
Current count of Media Packets Sent = 236
Current count of Media Packets Received = 236
Current count of RTCP Packets Sent = 0
Current count of RTCP Packets Received = 0
Average Call Duration = 22004
Average of the Answer Seizure Ratio = 1000
Average of the Round Trip Delay = 0 ms
Average of the locally calculated jitter = 0 ms
Average of the remotely calculated jitter = 0 ms
Average of the received media dropped per thousand pkts = 0
Average of the sent media lost per thousand pkts = 0

CAC Statistics

The CAC-related statistics have been enhanced to include the rejection counts for the CAC policies that have been implemented but failed.

A limiting field is configured in the CAC policy table entries and the CAC policy fails when there is a breach of that limit.

The following commands are used for displaying and clearing the CAC policy sets:

- The `show sbc name sbe cac-policy-set [id [table name [entry id]] | global [table name [entry id]]] [detail]` command—Lists information pertaining to the rejection counts for the failed CAC policies.
- The `clear sbc sbe-name sbc-policy-set-stats [all | policy-set cac-policy-number]` command—Clears all CAC policy statistics or clears statistics pertaining to a specified CAC policy set.

The following example shows how the `show sbc sbc-name sbc-cac-policy-set table entry` command displays statistic pertaining to the rejection counts for the failed CAC policies:

```
Router# show sbc sbc2 sbc-cac-policy-set 1 table table entry 1

SBC Service "SBC2"
CAC Averaging period 1: 60 sec
CAC Averaging period 2: 0 sec

CAC Policy Set 1
Global policy set: Yes
Description:
First CAC table: table
First CAC scope: global

Table name: table
Description:
Table type: policy-set
Total call setup failures (due to non-media limits): 0
```
### Entry 1

**CAC scope:**
CAC scope prefix length: 0

**Action:** CAC complete

**Number of call setup failures (due to non-media limits):** 0
**No. of registrations rejected (due to registration limits):** 0

**Max calls per scope:** Unlimited
**No. of events rejected due to Max Call Limit:** 0

**Max reg. per scope:** Unlimited
**No. of events rejected due to Max Reg limit:** 0

**Max channels per scope:** Unlimited
**Max updates per scope:** Unlimited
**Max bandwidth per scope:** Unlimited

#### Averaging-period 1

<table>
<thead>
<tr>
<th>Metric</th>
<th>Value 1</th>
<th>Value 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max call rate per scope</td>
<td>Unlimited</td>
<td>Unlimited</td>
</tr>
<tr>
<td>No. of events rejected due to Max call rate</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Max reg. rate per scope</td>
<td>Unlimited</td>
<td>Unlimited</td>
</tr>
<tr>
<td>No. of events rejected due to Max reg rate</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Max in-call message rate</td>
<td>Unlimited</td>
<td>Unlimited</td>
</tr>
<tr>
<td>No. of events rejected due to Max in-call rate</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Max out-call message rate</td>
<td>Unlimited</td>
<td>Unlimited</td>
</tr>
<tr>
<td>No. of events rejected due to Max Out call rate</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Timestamp when the rejection counts were last reset:** 2011/03/07 04:38:24

**Early media:** Allowed
**Early media direction:** Both

**Early media timeout:** None
**Transcoder per scope:** Allowed

**Callee Bandwidth-Field:** None
**Caller Bandwidth-Field:** None

**Media bypass:** Allowed
**Asymmetric Payload Type:** Not Set

**Renegotiate Strategy:** Delta
**SRTP Transport:** Trusted-Only (by default)

**Caller hold setting:** Standard
**Callee hold setting:** Standard

**Caller limited-privacy-service:** Never hide identity
**Callee limited-privacy-service:** Never hide identity

**Caller privacy-service:** Not set
**Callee privacy-service:** Not set

**Caller edit-privacy-request:** Not set
**Callee edit-privacy-request:** Not set

**Caller edit-privacy-request sip strip:** Not set
**Callee edit-privacy-request sip strip:** Not set

**Caller edit-privacy-request sip insert:** Not set
**Callee edit-privacy-request sip insert:** Not set

**Caller voice QoS profile:** Default
**Callee voice QoS profile:** Default

**Caller video QoS profile:** Default
**Callee video QoS profile:** Default

**Caller sig QoS profile:** Default
**Callee sig QoS profile:** Default

**Caller inbound SDP policy:** None
**Callee inbound SDP policy:** None

**Caller outbound SDP policy:** None
Callee outbound SDP policy: None
SDP Media Profile : None
Caller Generic Stream : Default
Callee Generic Stream : Default
Caller media disabled: None
Callee media disabled: None
Caller unsigned secure media: Not Allowed
Callee unsigned secure media: Not Allowed
Caller response downgrade support: No
Callee response downgrade support: No
Caller retry rtp support: No
Callee retry rtp support: No
Resend sdp answer in 200ok: No
Caller tel-event payload type: Default
Callee tel-event payload type: Default
Media flag: None
Restrict codecs to list: Default
Restrict caller codecs to list: Default
Restrict callee codecs to list: Default
Codec preference list: Default
Caller Codec profile: None
Callee Codec profile: None
Caller media caps list: None
Callee media caps list: None
TCS extra codec list: None
Caller media-type: Inherit (default)
Callee media-type: Inherit (default)
Caller Media Bypass: Inherit (default)
Callee Media Bypass: Inherit (default)
Media Bypass Type: Not set
Callee local transfer support: Inherit (default)
Maximum Call Duration: 50
Caller SRTP support: Inherit (default)
Callee SRTP support: Inherit (default)
SRTP Interworking: Inherit (default)
SRTP media Interworking: Inherit (default)
Ims rx preliminary-aar: Disabled(default)
Ims media-service: None(default)
media bandwidth policing: Inherit(default)
Billing filter: Inherit(default)
Caller ptime: None (default)
Callee ptime: None (default)
Caller codec variant conversion: Disabled (default)
Callee codec variant conversion: Disabled (default)
Callee inband DTMF mode: Inherit (default)
Callee Port Range Tag: Inherit (default)
Callee Port Range Tag: Inherit (default)
Session refresh renegotiation: Inherit (default)
Subscriber Statistics

Subscriber-related statistics have been introduced to enable you to view and analyze information pertaining to the subscribers who are using the SBC. The `show sbc sbe` command has been enhanced to display these statistics.

The following are the features of subscriber-related statistics:

- The statistics include the subscribers who are currently registered with the SBC, that is, the subscribers stored in the SBC subscriber database. Therefore, only subscribers registered through a non-IMS Access adjacency, P-CSCF Access adjacency, or IPsec P-CSCF Access adjacency are included.

- The statistics include the individual access-side subscribers who register through the SBC. The significance of this feature is that if the SBC rewrites the register, the address of record forwarded to the registrar may be the same for two different access-side address of records. These two addresses of record are recorded as two subscribers in the statistics even though the registrar may consider this as a single subscriber. For example, if both sip:12345@192.0.2.22 and sip:12345@203.0.113.36 register through the SBC and the SBC rewrites them to sip:12345@example.registrar.com, the registrar treats this entry as a single subscriber. However, the SBC counts this entry as two subscribers.

- The statistics represent the number of registered subscribers and not the number of registered contacts. When a subscriber registers multiple contacts, that subscriber will be counted only against the source adjacency of the first contact who is registered. For example, if a subscriber registers Contact1 on Adjacency1 and then registers Contact2 on Adjacency2, the subscriber is counted only once in the global count of subscribers. The same subscriber is not included in the count of subscribers on Adjacency2. This holds true even if Contact1 expires and Contact2 remains active.

- The statistics include delegate registrations performed by the SBC.

- The statistics are available both globally and per adjacency.

- In the IMS mode, the SBC may store multiple addresses of record for a single subscriber, for example, if the registrar returns P-Associated-URIs on a REGISTER response. In this scenario, only a single subscriber is included in the statistics and not the additional addresses of record.

Restrictions for the Subscriber Statistics Feature

The following are the restrictions for the subscriber statistics feature:

- The statistics are lost in the event of a failover.

- The statistics do not include the number of fast-registered subscribers.

- If subscribers register through the SBC on an Interconnection Border Control Function (IBCF) or non-IMS adjacency, the SBC does not track these subscribers and they are, therefore, not included in the statistics.

The following command displays subscriber-related statistics:

```
show sbc sbc-name sbe subscriber-stats { all | dst-account name | dst-adjacency name | global | src-account name | src-adjacency name | [current15mins | current5mins | currentday | currenthour | currentindefinite | previous15mins | previous5mins | previousday | previoushour ]
```

The following command resets all call-related statistics:

```
clear sbc sbc-name sbe call-stats { all | dst-account name | dst-adjacency name | global | src-account name | src-adjacency name } { all | currentindefinite }
```
The following example shows how the `show sbc sbe subscriber-stats` command displays statistics pertaining to subscribers:

```
Router# show sbc mySbc sbe subscriber-stats global currentindefinite

Subscribe count totals:
Active subscribers     = 10
Subscriber high water mark = 15
Subscriber low water mark = 3

Stats Reset Timestamp:
Timestamp when stats for this summary period were reset   = 2011/01/25 23:26:03
```

**Administrative Domains**

Each administrative domain represents administrative relationships with other peer entities, and can direct the SBC to use a particular number analysis and routing policy, and/or CAC policy for calls to and from the adjacency. The administrative domain is specified in admin-domain field for both SIP adjacencies and H.323 adjacencies. Any adjacencies without an administrative domain use globally configured policies.

An administrative domain has the following features:

- An administrative domain can identify policy trees that can be used for inbound or outbound number analysis, or taking a routing decision. These trees have all the attributes and capabilities of the existing number analysis and routing policy trees. The administrative domain can also identify zero or more CAC policy trees that have all the attributes and capabilities of the existing CAC policy tree.
- Users can create multiple separate policy trees for inbound number analysis, outbound number analysis, routing, and CAC. Each policy tree can be assigned to zero or more administrative domains. The user can bring each policy tree into and out of service independently from the others.
- An administrative domain can be identified by a text-based string that conveys the identity and scope of the domain.
- A signaling event can be assigned to a global administrative domain as its source or destination domain, if the classification system fails to assign it to any other source or destination administrative domain.
- Users can also assign a signaling message to multiple source and destination administrative domains. Each administrative domain is given a priority when it is assigned to an event. As per the priority given, the SBC uses the policy tree from the set of administrative domains.
- The user can assign a policy tree to administrative domain to take a routing decision for a signaling event. The user can also assign a policy tree to an administrative domain for outbound number analysis for the destination administrative domain. Changes to the policy trees can be made independently of each other. A routing decision is taken based on the policy tree chosen for the source administrative domain, but the outbound number analysis is based on the policy tree chosen for the destination administrative domain.
- All the source and destination administrative domains selected for a signaling event is provided to Billing Manager on detection points relating to that event. The XML format includes the names of the source and destination administrative domains in the billing record for a given call.
Asymmetric Payload Types

In Real-Time Transport Protocol (RTP) sessions, each codec is assigned an ID or payload type that is included in the RTP header. These payload types allow an RTP session to carry multiple formats, which may be different, concurrently. Different payload types can be assigned to the same codec in an RTP session.

If a session uses different payload types for the same codec, the session is said to be using asymmetric payload types.

SIP, H.323, and H.248 support asymmetric payload types. A SIP session negotiates the asymmetric payload types in RFC3264 Offer and Answer messages, while H.323 session negotiates the asymmetric payload types in the following messages:

- Fast Start request and response
- Open logical channel (OLC) and Open Logical Channel Acknowledgement
- Terminal Capabilities Set (applicable only to telephone-event codec)

The SBC enables seamless pass-through of asymmetric payload types in both signaling relay and media relay. Hence, an SBC can be used between two endpoints that use asymmetric payload types, without affecting the normal operations of the endpoints.

Asymmetric payload types are meant for only pass-through, and not for interworking. The SBC is not required to translate between asymmetric payload types on one leg of a call and symmetric payload types on the other leg of a call.

Prior to Cisco IOS XE Release 3.1.0S, if a SIP peer requested an asymmetric payload type, the SBC removed the codec that used the Asymmetric payload types. If no codecs were left, the entire call was torn down, as shown in Figure 7-8. From Cisco IOS XE Release 3.1.0S, the scenario illustrated in Figure 7-8 results in a successful call.
Figure 7-8  Asymmetric Payload Types —Call Teardown Scenarios

Caller

PT=a, Codec A
PT=b, Codec B

SBC

Codec B is removed because
Asymmetric Payload Types
are not supported

PT=a, Codec A

SBC does not support
Asymmetric Payload Types.
Hence, no codecs are
supported and the call is torn down.

Call teardown

Callee

PT=a, Codec A
PT=b, Codec B

Alternative 1
Single Asymmetric Payload Type

PT=a, Codec A
PT=c, Codec B

Alternative 2
Asymmetric Payload Types
for all codecs

PT=d, Codec A
PT=c, Codec B

Call teardown
Figure 7-9 shows a scenario where two codecs are present in an Offer, but only one is present—but twice—in the Answer. In this scenario, a combination of Asymmetric payload types and a changed codec is present.

**Figure 7-9   Asymmetric Payload Types—Two Codecs in Offer and One Codec in Answer**

In the example illustrated in Figure 7-9, the SBC matches the Answer to the Offer, mapping the first codecs together as using Asymmetric payload types, and then discards the second set of codecs as an unsupported codec change.

**Signaling**

This section describes how the Asymmetric payload types feature works in the following scenarios:

- SIP-RFC3264 Offer-Answer
- H323-H245
- H.323-SIP Interworking
- Media Programming
SIP-RFC3264 Offer-Answer

This section provides details on how Asymmetric payload types works in a SIP-RFC3264 Offer-Answer scenario.

Asymmetric Payload Types:
- Communicate payload type reassignment from Answer onwards to the offerer.
- Communicate to the MEDIA asymmetric payload type bindings negotiated by RFC3264.
- Log an event when they detect an answer that changes corresponding payload type in a media relay call.

When signaling originates a codec in an offer—either the telephone-event codec in dual tone multifrequency (DTMF) interworking, or a transcoder codec in transcoding—signaling accepts a change of payload type on the answer.

Refer to RFC3264 for more details on how payload type bindings are assigned by SDP rtpmap line.

H323-H245

For H323 calls, asymmetric payload types support is available only for the telephone-event codec.

The Asymmetric Payload Type feature affects only the processing of H.245 Terminal Capability Set, in particular, the receiveRTPAudioTelephonyEventCapability, which signals the RFC2833 telephone-event payload type. This feature:
- Communicates to MEDIA the asymmetric telephone-event payload type bindings negotiated by the H.245 Terminal Capability Set.
- Generates logs when it detects a different telephone-event payload type in each direction.

H.323-SIP Interworking

For H.323-SIP interworking, asymmetric payload types support is available only for the telephone-event codec. The telephone-event payload type received in an H.245 Terminal Capability Set message is communicated onwards in an RFC3264 Offer or Answer message and vice versa.

Although H.323 does not support Asymmetric payload types for any codec other than telephone-event, the same restriction does not apply to a SIP. Hence, a SIP peer might attempt to change the payload type on a flow as part of a SIP-H.323 interworking call. If the payload type is changed, a high-severity Problem Determination log is created, and the call is discarded.

Media Programming

Signaling uses standard H.248 signaling to program asymmetric payload type streams. During the transitions between a Symmetric and Asymmetric payload type bindings, media addresses or ports are not reallocated.

Billing

The Asymmetric Payload Types feature provides the following information for billing:
- Asymmetric payload types that are in use for a given media relay call.
- Codecs that are bound to each payload type.
Asymmetric Payload Types

SIP-SIP Calls

SIP calls indicate Asymmetric payload types by indicating differing payload types in an answer to the previous offer. The SBC will then act upon these Asymmetric payload types.

See the Example: Allowing Asymmetric Payload Types section on page 7-164 for examples of SIP/SIP configuration and Offer-Answer messages.

Configuring Asymmetric Payload Types

You can configure the SBC to allow or block Asymmetric payload types for each call. By default, asymmetric payload types are allowed on calls.

Use the payload-type asymmetric {allowed | disallowed} command to specify whether to allow or disallow asymmetric payload types.

Performing ISSU for Asymmetric Payload Types

When performing ISSU to upgrade to Cisco IOS XE Release 3.1.0S, a call requesting Asymmetric payload types from an active SBC with a release prior to Cisco IOS XE Release 3.1.0S is replicated to a standby with Cisco IOS XE Release 3.1.0S that supports Asymmetric payload types as if Symmetric Payload Types are being used. The media may not flow correctly on the primary or the backup after the failover.

If a call is currently using Symmetric payload types on an active SBC that does not support Asymmetric payload types, during attempts to renegotiate using Asymmetric payload types, one of the following occurs:

- If the Media, media forwarding component, or the Media Gateway detect that the active SBC does not support Asymmetric payload types, then the change to the corresponding call may be rejected and the call will remain unchanged.
- If the Media, media forwarding component, or Media Gateway does not detect that the active SBC does not support Asymmetric payload types, the corresponding call may continue as if using Symmetric payload types, and this may result in media not flowing correctly.

If a call changes the payload type from Symmetric to Asymmetric, or vice versa:

- After a gate is defined as Asymmetric, it remains Asymmetric even if it ceases to use Asymmetric payload types as a result of a renegotiation.
- If a Symmetric gate is marked as Asymmetric, and the partner does not support Asymmetric payload types, the gate is no longer replicated. The gate is deleted from the backup partner.
How to Implement Policies

Cisco Unified Border Element (SP Edition) policies are configured and activated as described in the following sections:

- Configuring Number Analysis Tables
- Configuring Administrative Domain
- Configuring Default Call Policy Set
- Configuring Routing Tables
- Configuring Number Manipulation
- Configuring Hunting
- Configuring H.323 MultiARQ Hunting
- Configuring Call Admission Control Policy Sets, CAC Tables, and Global CAC Policy Sets
- Configuring Privacy Service
- Configuring Multiple SBC Media Bypass
- Configuring Common IP Address Media Bypass
- Activating a CAC Policy Set

Configuring Number Analysis Tables

This task configures a number analysis table. The types of number analysis configuration are described in the following sections:

- Configuring Number Validation
- Configuring Number Categorization
- Configuring Text Address Validation and Source Address Manipulation

Configuring Number Validation

This task configures number validation for a number analysis table.

SUMMARY STEPS

1. configure terminal
2. sbc sbc-name
3. sbe
4. call-policy-set policy-set-id
5. first-inbound-na-table table-name
6. na-dst-prefix-table table-name
7. entry entry-id
8. match-prefix key
9. action [next-table goto-table-name | accept | reject]
10. category category-name
How to Implement Policies

11. entry entry-id
12. edit [del-prefix pd] | [del-suffix sd] | [add-prefix pa] | [replace ds]
13. edit-cic [del-prefix pd] | [del-suffix sd] | [add-prefix pa] | [replace ds]
14. match-prefix key
15. action [next-table goto-table-name | accept | reject]
16. category category-name
17. entry entry-id
18. match-prefix key
19. action [next-table goto-table-name | accept | reject]
20. category category-name
21. exit
22. exit
23. end
24. show

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure terminal</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router# configure terminal</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>sbc sbc-name</td>
</tr>
</tbody>
</table>
| **Example:** | Router(config)# sbc mySbc  
Router(config-sbc)# | |
| **Step 3**  | sbe | Enters the mode of an SBE entity within an SBC service. |
| **Example:** | Router(config-sbc)# sbe  
Router(config-sbc-sbe)# | |
| **Step 4**  | call-policy-set policy-set-id | Enters the mode of routing policy set configuration within an SBE entity, creating a new policy set, if necessary. |
| **Example:** | Router(config-sbc-sbe)# call-policy-set 1  
Router(config-sbc-sbe-rtgpolicy)# | |
| **Step 5**  | first-inbound-na-table table-name | Configures the name of the first policy table to process when performing the number analysis stage of policy. |
| **Example:** | Router(config-sbc-sbe-rtgpolicy)#  
first-inbound-na-table hotel_table | |
### How to Implement Policies

#### Step 6

**Command or Action:**

```
na-dst-prefix-table table-name
```

**Purpose:**
Enters the mode for configuring a number analysis table whose entries match the prefix (the first several digits) of the dialed number within the context of an SBE policy set.

**Example:**

```
Router(config-sbc-sbe-rtgpolicy)#
na-dst-prefix-table hotel_table
```

#### Step 7

**Command or Action:**

```
entry entry-id
```

**Purpose:**
Enters the mode for configuring an entry in a number analysis table, creating the entry, if necessary.

**Example:**

```
Router(config-sbc-sbe-rtgpolicy-natable)# entry 1
```

#### Step 8

**Command or Action:**

```
match-prefix key | match-cic cic
```

**Purpose:**
Configures the match value of an entry in the number analysis table.

**Example:**

```
Router(config-sbc-sbe-rtgpolicy-natable-entry)# match-prefix XXX
```

- The `match-prefix key` argument is a string used to match the prefix (the starting part) of the dialed number.
- The `match-cic cic` argument is used with the `na-carrier-id-table` command and configures the match carrier ID code in a table whose entries match a carrier ID.

#### Step 9

**Command or Action:**

```
action [next-table goto-table-name | accept | reject]
```

**Purpose:**
Configures the action of an entry in a number analysis table. Possible actions are:

- Configure the name of the next number analysis table to process if the event matches this entry using the `next-table` keyword and the `goto-table-name` argument.
- Configure the call to be accepted if it matches the entry in the table using the `accept` keyword.
- Configure the call to be rejected if it matches the entry in the table using the `reject` keyword.

**Example:**

```
Router(config-sbc-sbe-rtgpolicy-natable-entry)# action accept
```

#### Step 10

**Command or Action:**

```
category category-name
```

**Purpose:**
Configures the category of an entry in the number analysis table.

**Example:**

```
Router(config-sbc-sbe-rtgpolicy-natable-entry)# category external
```
How to Implement Policies

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 11</strong></td>
<td></td>
</tr>
<tr>
<td>entry entry-id</td>
<td>Enters the mode for configuring an entry in a number analysis table, creating the entry, if necessary.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-rtgpolicy-natable-entry)#</td>
<td></td>
</tr>
<tr>
<td>entry 2</td>
<td></td>
</tr>
<tr>
<td><strong>Step 12</strong></td>
<td></td>
</tr>
</tbody>
</table>
| edit [del-prefix pd] | [del-suffix sd] | [add-prefix pa] | [replace ds] | Configures a dial-string manipulation action in a number analysis table. You are not allowed to do this if the table is part of the active policy set. The no version of the command deletes the edit action of the given entry in the routing table. The edit command can be set to the following values:
- **del-prefix pd**—Delete prefix pd, where pd is a positive integer specifying a number of digits to delete from the front of the dialed string.
- **del-suffix sd**—Delete suffix sd, where sd is a positive integer specifying a number of digits to delete from the end of the dialed string.
- **add-prefix pa**—Add prefix pa, where pa is a string of digits to add to the front of the dialed string.
- **replace ds**—Replace ds, where ds is a string of digits that replaces the dialed string.

In the example to the left, the edit command sets entry 2 to delete 1 digit from the beginning of the dialed string in the number analysis table.

| Example:                                                |                                                                          |
|                                                        |                                                                          |
| Router(config-sbc-sbe-rtgpolicy-natable-entry)#        |                                                                          |
| edit del-prefix 1                                       |                                                                          |
### Step 13
**edit-cic [del-prefix pd] | [del-suffix sd] | [add-prefix pa] | [replace ds]**

**Example:**
Router(config-sbc-sbe-rtgpolicy-natable-entry)#
edit-cic del-prefix 1

Configures a carrier identification code (CIC) manipulation action in a number analysis table. You are not allowed to do this if the table is part of the active policy set.

- **del-prefix pd**: A positive integer specifying a number of digits to delete from the front of the carrier ID string.
- **del-suffix sd**: A positive integer specifying a number of digits to delete from the end of the carrier ID string.
- **add-prefix pa**: A string of digits to add to the front of the carrier ID string.
- **replace ds**: A string of digits to replace the carrier ID string with.

The "edit-cic del-prefix 1" command sets entry 2 to delete the first digit of the carrier ID in the current number analysis table.

You can remove the CIC or carrier ID from outbound messages by specifying a replacement string of 0000 or by specifying a prefix deletion length of 4.

For example:
```
edit-cic del-prefix 4   OR
edit-cic replace 0000
```

### Step 14
**match-prefix key**

**Example:**
Router(config-sbc-sbe-rtgpolicy-natable-entry)#
match-prefix 9XXX

Configures the match value of an entry in the number analysis table. The key argument is a string used to match the start of the dialed number.

The no version of the command destroys the match value.

### Step 15
**action [next-table goto-table-name] | accept | reject**

**Example:**
Router(config-sbc-sbe-rtgpolicy-natable-entry)#
action accept

Configures the action of an entry in a number analysis table. Possible actions are:

- Configure the name of the next number analysis table to process if the event matches this entry using the **next-table** keyword and the **goto-table-name** argument.
- Configure the call to be accepted if it matches the entry in the table using the **accept** keyword.
- Configure the call to be rejected if it matches the entry in the table using the **reject** keyword.

### Step 16
**category category-name**

**Example:**
Router(config-sbc-sbe-rtgpolicy-natable-entry)#
category external

Configures the category of an entry in the number analysis table.
### How to Implement Policies

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 17</strong> entry entry-id</td>
<td>Enters the mode for configuring an entry in a number analysis table, creating the entry, if necessary.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-rtgpolicy-natable-entry)# entry 3</td>
<td></td>
</tr>
<tr>
<td><strong>Step 18</strong> match-prefix key</td>
<td>Configures the match value of an entry in the number analysis table. The key argument is a string used to match the start of the dialed number.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-rtgpolicy-natable-entry)# match-prefix 8XXX</td>
<td></td>
</tr>
<tr>
<td><strong>Step 19</strong> action [next-table goto-table-name</td>
<td>accept</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-rtgpolicy-natable-entry)# action accept</td>
<td></td>
</tr>
<tr>
<td><strong>Step 20</strong> category category-name</td>
<td>Configures the category of an entry in the number analysis table.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-rtgpolicy-natable-entry)# category bar</td>
<td></td>
</tr>
<tr>
<td><strong>Step 21</strong> exit</td>
<td>Exits from the entry mode to the natable mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-rtgpolicy-natable-entry)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 22</strong> exit</td>
<td>Exits from the natable mode to the callpolicy mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-rtgpolicy-natable)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 23</strong> end</td>
<td>Exits the callpolicy mode to Privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-rtgpolicy-natable)# end</td>
<td></td>
</tr>
<tr>
<td><strong>Step 24</strong> show</td>
<td>Displays the current configuration information.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-rtgpolicy)# show</td>
<td></td>
</tr>
</tbody>
</table>
Configuring Number Categorization

This task configures number categorization for a number analysis table.

**SUMMARY STEPS**

1. configure terminal
2. sbc sbc-name
3. sbc
4. call-policy-set policy-set-id
5. first-inbound-na-table table-name
6. na-src-account-table table-name
7. entry entry-id
8. match-account key
9. action [next-table goto-table-name | accept | reject]
10. entry entry-id
11. match-account key
12. action [next-table goto-table-name | accept | reject]
13. entry entry-id
14. match-account key
15. action [next-table goto-table-name | accept | reject]
16. na-dst-prefix-table table-name
17. entry entry-id
18. match-prefix key
19. category category-name
20. action [next-table goto-table-name | accept | reject]
21. entry entry-id
22. match-prefix key
23. category category-name
24. action [next-table goto-table-name | accept | reject]
25. end
26. show
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enables the global configuration mode.</td>
</tr>
</tbody>
</table>
| **Step 2** sbc sbc-name | Enters the SBC service mode.  
  - sbc-name—Name of the SBC. |
<p>| <strong>Step 3</strong> sbe | Enters the mode of an SBE entity within an SBC service. |
| <strong>Step 4</strong> call-policy-set policy-set-id | Enters the mode of routing policy set configuration within an SBE entity, creating a new policy set if necessary. |
| <strong>Step 5</strong> first-inbound-na-table table-name | Configures the name of the first policy table to process when performing the number analysis stage of policy. |
| <strong>Step 6</strong> na-src-account-table table-name | Enters the mode for configuring a number analysis table within the context of an SBE policy set with the entries of the table matching the source account. |
| <strong>Step 7</strong> entry entry-id | Enters the mode for configuring an entry in a number analysis table, creating the entry, if necessary. |
| <strong>Step 8</strong> match-account key | Configures the match value of an entry in the number analysis table. The key argument is a string used to match the source account. |</p>
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 9</strong></td>
<td>**action [next-table goto-table-name</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-rtgpolicy-natable-entry)# action next-table hotel_dialing_plan</td>
</tr>
<tr>
<td><strong>Purpose</strong></td>
<td>Configures the action of an entry in a number analysis table. Possible actions are:</td>
</tr>
<tr>
<td></td>
<td>• Configure the name of the next number analysis table to process if the event matches this entry using the <strong>next-table</strong> keyword and the <strong>goto-table-name</strong> argument.</td>
</tr>
<tr>
<td></td>
<td>• Configure the call to be accepted if it matches the entry in the table using the <strong>accept</strong> keyword.</td>
</tr>
<tr>
<td></td>
<td>• Configure the call to be rejected if it matches the entry in the table using the <strong>reject</strong> keyword.</td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td><strong>entry entry-id</strong></td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-rtgpolicy-natable-entry)# entry 2</td>
</tr>
<tr>
<td><strong>Purpose</strong></td>
<td>Enters the mode for configuring an entry in a number analysis table, creating the entry, if necessary.</td>
</tr>
<tr>
<td><strong>Step 11</strong></td>
<td><strong>match-account key</strong></td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-rtgpolicy-natable-entry)# match-account hotel_bar</td>
</tr>
<tr>
<td><strong>Purpose</strong></td>
<td>Configures the match value of an entry in the number analysis table. The <strong>key</strong> argument is a string used to match the source account.</td>
</tr>
<tr>
<td><strong>Step 12</strong></td>
<td>**action [next-table goto-table-name</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-rtgpolicy-natable-entry)# action next-table hotel_dialing_plan</td>
</tr>
<tr>
<td><strong>Purpose</strong></td>
<td>Configures the action of an entry in a number analysis table. Possible actions are:</td>
</tr>
<tr>
<td></td>
<td>• Configure the name of the next number analysis table to process if the event matches this entry using the <strong>next-table</strong> keyword and the <strong>goto-table-name</strong> argument.</td>
</tr>
<tr>
<td></td>
<td>• Configure the call to be accepted if it matches the entry in the table using the <strong>accept</strong> keyword.</td>
</tr>
<tr>
<td></td>
<td>• Configure the call to be rejected if it matches the entry in the table using the <strong>reject</strong> keyword.</td>
</tr>
<tr>
<td><strong>Step 13</strong></td>
<td><strong>entry entry-id</strong></td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-rtgpolicy-natable-entry)# entry 3</td>
</tr>
<tr>
<td><strong>Purpose</strong></td>
<td>Enters the mode for configuring an entry in a number analysis table, creating the entry, if necessary.</td>
</tr>
<tr>
<td><strong>Step 14</strong></td>
<td><strong>match-account key</strong></td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-rtgpolicy-natable-entry)# match-account internal</td>
</tr>
<tr>
<td><strong>Purpose</strong></td>
<td>Configures the match value of an entry in the number analysis table. The <strong>key</strong> argument is a string used to match the source account.</td>
</tr>
</tbody>
</table>
## How to Implement Policies

### Command or Action | Purpose
--- | ---
**Step 15** action [next-table goto-table-name | accept | reject] | Configures the action of an entry in a number analysis table. Possible actions are:
- Configure the name of the next number analysis table to process if the event matches this entry using the **next-table** keyword and the **goto-table-name** argument.
- Configure the call to be accepted if it matches the entry in the table using the **accept** keyword.
- Configure the call to be rejected if it matches the entry in the table using the **reject** keyword.

**Example:**
```
Router(config-sbc-sbe-rtgpolicy-natable-entry)# action accept
```

**Step 16** na-dst-prefix-table table-name | Enters the mode for configuring a number analysis table within the context of an SBE policy set with the entries of the table matching the start of the dialed number.

**Example:**
```
Router(config-sbc-sbe-rtgpolicy-natable-entry)# na-dst-prefix-table hotel_dialing_plan
```

**Step 17** entry entry-id | Enters the mode for configuring an entry in a number analysis table, creating the entry, if necessary.

**Example:**
```
Router(config-sbc-sbe-rtgpolicy-natable-entry)# entry 1
```

**Step 18** match-prefix key | Configures the match value of an entry in the number analysis table. The **key** argument is a string used to match the start of the dialed number.

**Example:**
```
Router(config-sbc-sbe-rtgpolicy-natable-entry)# match-prefix XXX
```

**Step 19** category category-name | Specifies the category of an entry in a number analysis table.

**Example:**
```
Router(config-sbc-sbe-rtgpolicy-natable-entry)# category internal_call
```

**Step 20** action [next-table goto-table-name | accept | reject] | Configures the action of an entry in a number analysis table. Possible actions are:
- Configure the name of the next number analysis table to process if the event matches this entry using the **next-table** keyword and the **goto-table-name** argument.
- Configure the call to be accepted if it matches the entry in the table using the **accept** keyword.
- Configure the call to be rejected if it matches the entry in the table using the **reject** keyword.

**Example:**
```
Router(config-sbc-sbe-rtgpolicy-natable-entry)# action accept
```

**Step 21** entry entry-id | Enters the mode for configuring an entry in a number analysis table, creating the entry, if necessary.

**Example:**
```
Router(config-sbc-sbe-rtgpolicy-natable-entry)# entry 2
```
How to Implement Policies

Configuring Text Address Validation and Source Address Manipulation

This task shows how to configure text address validation and source address manipulation for a number analysis table.

**SUMMARY STEPS**

1. configure terminal
2. sbc sbc-name
3. sbe
4. call-policy-set policy-set-id
5. first-inbound-na-table table-name
6. na-dst-address-table table-name
7. entry entry-id
8. action [next-table goto-table-name | accept | reject]
Chapter 7 Implementing Cisco Unified Border Element (SP Edition) Policies

How to Implement Policies

9. edit-src [del-prefix pd] | [del-suffix sd] | [add-prefix pa] | [replace ds]
10. match-address key [regex | digits]
11. entry entry-id
12. action [next-table goto-table-name | accept | reject]
13. edit-src [del-prefix pd] | [del-suffix sd] | [add-prefix pa] | [replace ds]
14. match-address key [regex | digits]
15. exit
16. exit
17. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure terminal</td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 2 sbc sbc-name</td>
<td>Enters the SBC service mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config)# sbc mySbc</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc)#</td>
<td></td>
</tr>
<tr>
<td>Use the sbc-name argument</td>
<td></td>
</tr>
<tr>
<td>Step 3 sbe</td>
<td>Enters the SBE entity mode within an SBC service.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe)#</td>
<td></td>
</tr>
<tr>
<td>Step 4 call-policy-set</td>
<td>Enters the routing policy set configuration mode within an SBE entity,</td>
</tr>
<tr>
<td>policy-set-id</td>
<td>creating a new policy set, if necessary.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe)#</td>
<td></td>
</tr>
<tr>
<td>call-policy-set 1</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-rtgpolicy)#</td>
<td></td>
</tr>
<tr>
<td>Step 5 first-inbound-na-table</td>
<td>Configures the name of the first policy table to be processed</td>
</tr>
<tr>
<td>table-name</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-rtgpolicy)#</td>
<td></td>
</tr>
<tr>
<td>first-inbound-na-table</td>
<td></td>
</tr>
<tr>
<td>hotel_table</td>
<td></td>
</tr>
</tbody>
</table>
Step 6  
**na-dst-address-table** *table-name*  

**Example:**  
Router(config-sbc-sbe-rtgpolicy)# na-dst-address-table room_table  

Enters the number analysis table mode for configuring a number analysis table whose entries match the prefix (the first few digits) of the dialed number within the context of an SBE policy set.  

The commands for other number analysis tables are:  
- na-carrier-id-table (This table requires additional commands—**match-cic** and **edit-cic**)  
- na-dst-address-table  
- na-src-address-table  
- na-src-prefix-table  
- na-src-account-table  
- na-src-adjacency-table  
- na-carrier-id-table  

Step 7  
**entry** *entry-id*  

**Example:**  
Router(config-sbc-sbe-rtgpolicy-natable)# entry 1  

Enters the number analysis table entry mode for configuring an entry in a number analysis table, creating the entry, if necessary.  

Step 8  
**action** [**next-table** *goto-table-name* | **accept** | **reject**]  

**Example:**  
Router(config-sbc-sbe-rtgpolicy-natable-entry)# action accept  

Configures the action of an entry in a number analysis table. Possible actions are:  
- Configure the name of the next number analysis table to be processed if the event matches this entry, using the **next-table** keyword and the **goto-table-name** argument.  
- Configure the call to be accepted if it matches the entry in the table, using the **accept** keyword.  
- Configure the call to be rejected if it matches the entry in the table, using the **reject** keyword.  

Step 9  
**edit-src** [**del-prefix** *pd*] | [**del-suffix** *sd*] | [**add-prefix** *pa*] | [**replace** *ds*]  

**Example:**  
Router(config-sbc-sbe-rtgpolicy-natable-entry)# edit-src del-prefix 3  

Configures the source address manipulation action in the NA table. This cannot be done if a table is part of the active policy set. The **no** version of the command removes the match value.  
- **del-prefix** *pd*: A positive integer specifying the number of digits to be delete from the front of the carrier ID string.  
- **del-suffix** *sd*: A positive integer specifying the number of digits to be deleted from the end of the carrier ID string.  
- **add-prefix** *pa*: A string of digits to be added to the front of the carrier ID string.  
- **replace** *ds*: A string of digits to replace the carrier ID string with.
How to Implement Policies

### Step 10
**match-address** key [regex | digits]

**Example:**
```bash
Router(config-sbc-sbe-rtgpolicy-natable-entry)#
mMatch-address 123456 digits
```

Confirms the match value of an entry in an NA table.

To create a routing table that routes on user name, use the existing rtg-dst-address-table or rtg-src-address-table, and include a textual value in the match-address field.

The SBC skips number analysis and performs only routing when the SIP message contains a user name. The SBC decides that an address is a user name (as opposed to a phone number) if the address contains any character other than 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, plus, hyphen, period, open-round-bracket, and close-round-bracket.

When the SBC has decided that an address is a user name, the X in the routing tables is treated not as a wildcard character, but as a literal X. For example, the match value of X matches the username X, but not A.

**Note** A direct string comparison is not done by NA. To compare a fixed string, a regex without any regex meta-characters can be used.

### Step 11
**entry** entry-id

**Example:**
```bash
Router(config-sbc-sbe-rtgpolicy-natable-entry)#
entry 2
```

Enters the number analysis table entry mode for configuring an entry in a number analysis table, creating the entry, if necessary.

### Step 12
**action** [next-table goto-table-name | accept | reject]

**Example:**
```bash
Router(config-sbc-sbe-rtgpolicy-natable-entry)#
action accept
```

Configures the action of an entry in a number analysis table. Possible actions are:

- Configure the name of the next number analysis table to be processed if the event matches this entry, using the **next-table** keyword and the **goto-table-name** argument.
- Configure the call to be accepted if it matches the entry in the table, using the **accept** keyword.
- Configure the call to be rejected if it matches the entry in the table, using the **reject** keyword.
### Command or Action

**Step 13**

```plaintext
edit-src | [del-prefix pd] | [del-suffix sd] | [add-prefix pa] | [replace ds]
```

**Example:**

Router(config-sbc-sbe-rtgpolicy-natable-entry)#

```plaintext
edit-src del-suffix 1
```

**Purpose:**

Configures the source address manipulation action in the NA table.

This cannot be done if the table is a part of the active policy set.

The **no** version of the command destroys the match value.

- **del-prefix pd:** A positive integer specifying the number of digits to be deleted from the front of the carrier ID string.
- **del-suffix sd:** A positive integer specifying the number of digits to be deleted from the end of the carrier ID string.
- **add-prefix pa:** A string of digits to be added to the front of the carrier ID string.
- **replace ds:** A string of digits to be replaced the carrier ID string with.

**Step 14**

```plaintext
match-address key [regex | digits]
```

**Example:**

Router(config-sbc-sbe-rtgpolicy-natable-entry)#

```plaintext
match-address ^.* regex
```

**Purpose:**

Configures the match value of an entry in an NA table.

To create a routing table that routes on user name, use the existing rtg-dst-address-table or rtg-src-address-table, and include a textual value in the match-address field.

The SBC skips number analysis and performs only routing when the SIP message contains a user name. The SBC decides that an address is a user name (as opposed to a phone number) if the address contains any character other than 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, plus, hyphen, period, open-round-bracket, and close-round-bracket.

When the SBC has decided that an address is a user name, the X in the routing tables is treated not as a wildcard character, but as a literal X. For example, the match value of X matches the username X, but not A.

**Note**

A direct string comparison is not done by NA. To compare a fixed string, a regex without any regex meta-characters can be used.

**Step 15**

```plaintext
exit
```

**Example:**

Router(config-sbc-sbe-rtgpolicy-natable-entry)#

```plaintext
exit
```

**Purpose:**

Exits from the entry mode and enters the natable mode.

**Step 16**

```plaintext
exit
```

**Example:**

Router(config-sbc-sbe-rtgpolicy-natable)#

```plaintext
exit
```

**Purpose:**

Exits from the natable mode and enters the call policy mode.

**Step 17**

```plaintext
end
```

**Example:**

Router(config-sbc-sbe-rtgpolicy)#

```plaintext
end
```

**Purpose:**

Exits the call policy mode and enters the Privileged EXEC mode.
Configuring Administrative Domain

This task configures an administrative domain.

**Note**

The policy sets must be in a complete state before they are assigned to an administrative domain. A default call-policy-set must be configured before the administrative domain mode is entered. If an inbound NA set, a routing set, or an outbound NA set is undefined, the administrative domain uses the values defined within the default call-policy-set.

**SUMMARY STEPS**

1. `configure terminal`
2. `sbc sbc-name`
3. `sbe`
4. `admin-domain name`
5. `description [line]`
6. `call-policy-set {inbound-na number | outbound-na number | rtg number} [priority priority-value]`
7. `cac-policy-set number`
8. `exit`
9. `adjacency sip | h323 adjacency-name`
10. `admin-domain name`
11. `end`
12. `show sbc sbc-name sbe admin-domain [adjacency]`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 <code>configure terminal</code></td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 2 <code>sbc sbc-name</code></td>
<td>Enters the mode of an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# sbc mySbc</td>
<td></td>
</tr>
<tr>
<td>Step 3 <code>sbe</code></td>
<td>Enters the mode of an SBE entity within an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td>Step 4 <code>admin-domain name</code></td>
<td>Enters the mode of an administrative domain.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe)# admin-domain Domain1</td>
<td></td>
</tr>
</tbody>
</table>
### Step 5
**description [line]**

Assigns a text description to the administrative domain.
- *line*—Describes the administrative domain.

**Example:**
```
Router(config-sbc-sbe-ad)# description This is a description of DOMAIN1
```

### Step 6
**call-policy-set {inbound-na number | outbound-na number | rtg number} [priority priority-value]**

Configures a single call-policy-set or separate call-policy-sets for routing, inbound number analysis, and outbound number analysis. The policy sets must be in a complete state before they can be assigned to the policy set of an administrative domain.

**Note** Specifying an inbound NA, a routing, or an outbound NA policy set is optional. If the policy sets are undefined, the admin-domain uses the values defined within the default call policy set.

- **inbound-na**—Specifies the inbound number analysis policy
- **outbound-na**—Specifies the outbound number analysis policy
- **rtg**—Specifies the routing policy
- **priority**—Specifies the priority of a policy-set.
- **number**—An unique identifier for the policy set. The value can range from 1 to 2147483647.
- **priority-value**—The priority value ranging from 1 to 10 where 10 indicates the highest priority. By default, the priority is set to 10.

**Example:**
```
Router(config-sbc-sbe-ad)# call-policy-set rtg 2 priority 1
Router(config-sbc-sbe-ad)# call-policy-set inbound-na 2 priority 1
Router(config-sbc-sbe-ad)# call-policy-set outbound-na 2 priority 1
```

### Step 7
**cac-policy-set number**

Configures the cac-policy-set in an administrative domain. Only one cac-policy-set can be specified.

**Example:**
```
Router(config-sbc-sbe-ad)# cac-policy-set 2
```

**Note** Priority is required because more than one administrative domain can be specified on an adjacency. The SBC uses the policy-set with the highest priority.

### Step 8
**exit**

Exits the administrative domain mode and enters the SBE mode.

**Example:**
```
Router(config-sbc-sbe-ad)# exit
```

### Step 9
**adjacency sip | h323 adjacency-name**

Enters the mode of an SBE SIP or H.323 adjacency.

- **adjacency-name**— Defines the name of the SIP or H.323 adjacency.

**Note** The H323 adjacency must be unattached to add, delete, or modify the admin-domain command.
How to Implement Policies

The following example shows the output of the `show sbc sbe admin-domain` command:

```
Router# show sbc MySBC sbe admin-domain
SBC Service "MySBC"
Global cac-policy-set: 2
Default call-policy-set/priority: 1/6
```

<table>
<thead>
<tr>
<th>Administrative Domain</th>
<th>cac policy-set</th>
<th>call-policy-set/priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOMAIN1</td>
<td>2</td>
<td>2/1</td>
</tr>
</tbody>
</table>

The following example shows the output of the `show sbc sbe admin-domain adjacency` command:

```
Router# show sbc MySBC sbe admin-domain adjacency
SBC Service "MySBC"
```

<table>
<thead>
<tr>
<th>Adjacency Name</th>
<th>Type</th>
<th>State</th>
<th>Admin-domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIPP1A</td>
<td>SIP</td>
<td>Attached</td>
<td>DOMAIN1</td>
</tr>
</tbody>
</table>

Configuring Default Call Policy Set

This task configures a call-policy-set and sets a priority for the SBC to determine the default policy set to use when the administrative domain is not present.

SUMMARY STEPS

1. configure terminal
2. sbc sbc-name
3. sbe
4. call-policy-set policy-set-id
5. first-inbound-na-table word
6. first-outbound-na-table word
7. complete
8. `exit`
9. `call-policy-set default policy-set-id [priority priority]`
10. `end`
11. `show sbc sbc-name sbe call-policy [default]`

### Detailed Steps

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure terminal</td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 2 sbc sbc-name</td>
<td>Enters the mode of an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# sbc mySbc</td>
<td>• <code>sbc-name</code>—The name of the SBC service.</td>
</tr>
<tr>
<td>Step 3 sbe</td>
<td>Enters the mode of an SBE entity within an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td>Step 4 call-policy-set policy-set-id</td>
<td>Creates a new call policy set and enters SBE routing policy configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe)# call-policy-set 25</td>
<td>• <code>policy-set-id</code>—The call policy set number that can range from 1 to 2147483647.</td>
</tr>
<tr>
<td>Step 5 first-inbound-na-table word</td>
<td>Specifies the first inbound number analysis table.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-rtgpolicy)# first-inbound-na-table InTable</td>
<td>• <code>word</code>—Inbound number analysis table name. The table length can be of 30 characters maximum.</td>
</tr>
<tr>
<td>Step 6 first-outbound-na-table word</td>
<td>Specifies the first outbound number analysis table.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-rtgpolicy)# first-outbound-na-table InTable</td>
<td>• <code>word</code>—Outbound number analysis table name. The table length can be of 30 characters maximum.</td>
</tr>
<tr>
<td>Step 7 complete</td>
<td>Completes the call-policy set after committing the full set.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-rtgpolicy)# complete</td>
<td></td>
</tr>
<tr>
<td>Step 8 exit</td>
<td>Exits the SBE routing policy mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-rtgpolicy)# exit</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Step 9</th>
<th>call-policy-set default policy-set-id [priority priority]</th>
</tr>
</thead>
</table>

**Example:**

```
Router(config-sbc-sbe)# call-policy-set default 25 priority 1
```

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Assigns the default call-policy-set id when an administrative domain is not specified on the adjacency or the specified administrative domain does not exist.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• policy-set-id—The call policy set number, ranging from 1 to 2147483647. The policy set must be in a complete state before it can be assigned as the default policy.</td>
</tr>
<tr>
<td></td>
<td>• priority—Specifies the priority to determine which active call-policy-set to use. The SBC uses the policy set with the highest priority.</td>
</tr>
<tr>
<td></td>
<td>• priority—The priority value ranging from 1 to 10 with 10 indicating highest priority. By default, priority is set to 6.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> A default call-policy-set must be configured before the user enters the administrative domain mode. If an inbound NA set, a routing set, or an outbound NA set is undefined, the administrative domain uses the values defined within the default call-policy-set.</td>
</tr>
</tbody>
</table>

### Step 10

<table>
<thead>
<tr>
<th>end</th>
</tr>
</thead>
</table>

**Example:**

```
Router(config-sbc-sbe)# end
```

**Exits the SBE mode and enters the Privilege exec mode.**

### Step 11

<table>
<thead>
<tr>
<th>show sbe sbe-name sbe call-policy-set [default]</th>
</tr>
</thead>
</table>

**Example:**

```
Router# show sbe mySBC sbe call-policy-set
```

**Displays details of the call policy sets configured on the SBC.**

- **sbc-name**—Defines the name of the SBC service.
- **default**—Lists the information pertaining to the default call policy set.

The following example shows the output of the `show sbe sbe call-policy-set` command:

```
Router# show sbe mySBC sbe call-policy-set

SBC Service "mySBC"

Policy set 1
  Default policy set : Yes (priority 6)
  First inbound NA table :
  First call routing table : TAB1
  First reg routing table : TAB2
  First outbound NA table :

  Table Name : TAB1
  Class : Routing
  Table type : rtg-src-adj
  Total Call-policy Failures : 0 (0 *)

  Entry    Match Value    Destination Adjacency    Action
  -------  -------------  ------------------------  ----
  1      SIPP1A      SIPP1B                  Routing complete
  2      SIPP1B      SIPP1A                  Routing complete

  Table Name : TAB2
  Class : Routing
```
### Table: Call-policy Failures

<table>
<thead>
<tr>
<th>Entry</th>
<th>Match Value</th>
<th>Destination Adjacency</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SIPP1A</td>
<td>Registrar</td>
<td>Routing complete</td>
</tr>
<tr>
<td>2</td>
<td>SIPP1B</td>
<td>Registrar</td>
<td>Routing complete</td>
</tr>
</tbody>
</table>

### Policy set 2

**Default policy set**: No
**First inbound NA table**: 
**First call routing table**: TAB1
**First reg routing table**: TAB2
**First outbound NA table**: 

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Class</th>
<th>Table type</th>
<th>Total Call-policy Failures</th>
<th>Entry</th>
<th>Match Value</th>
<th>Destination Adjacency</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>TAB1</td>
<td>Routing</td>
<td>rtg-src-adj</td>
<td>0 (0 *)</td>
<td>1</td>
<td>SIPP1A</td>
<td>SIPP1B</td>
<td>Routing complete</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>SIPP1B</td>
<td>SIPP1A</td>
<td>Routing complete</td>
</tr>
</tbody>
</table>

### Policy set 25

**Default policy set**: No
**First inbound NA table**: ADMINTable
**First call routing table**: 
**First reg routing table**: 
**First outbound NA table**: OutTable

* Numbers in brackets refer to a call being rejected by a routing or number analysis table because there were no matching entries in the table. This is also included in the total figure.

The following example shows the output of the `show sbc sbe call-policy-set default` command:

```
Router# show sbc mySBC sbe call-policy-set default

SBC Service "mySBC"

Policy set 1
**Default policy set**: Yes (priority 6)
**First inbound NA table**: 
**First call routing table**: TAB1
**First reg routing table**: TAB2
**First outbound NA table**: 

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Class</th>
<th>Table type</th>
<th>Total Call-policy Failures</th>
<th>Entry</th>
<th>Match Value</th>
<th>Destination Adjacency</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>TAB1</td>
<td>Routing</td>
<td>rtg-src-adj</td>
<td>0 (0 *)</td>
<td>1</td>
<td>SIPP1A</td>
<td>SIPP1B</td>
<td>Routing complete</td>
</tr>
</tbody>
</table>

```
How to Implement Policies

2          SIPP1B               SIPP1A                 Routing complete

Table Name                   : TAB2
Class                      : Routing
Table type                 : rtg-src-adj
Total Call-policy Failures : 0 (0 *)

<table>
<thead>
<tr>
<th>Entry</th>
<th>Match Value</th>
<th>Destination Adjacency</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SIPP1A</td>
<td>Registrar</td>
<td>Routing complete</td>
</tr>
<tr>
<td>2</td>
<td>SIPP1B</td>
<td>Registrar</td>
<td>Routing complete</td>
</tr>
</tbody>
</table>

* Numbers in brackets refer to a call being rejected by a routing or number analysis table because there were no matching entries in the table. This is also included in the total figure.

Configuring Routing Tables

See the following sections:
- Configuring a Destination Address Table, page 7-86
- Configuring the Destination, Source Domain, and Carrier ID Tables, page 7-92
- Configuring Number Manipulation, page 7-104
- Configuring the Least Cost Table, page 7-97
- Configuring Time-Based Tables, page 7-99
- Configuring Trunk-Group ID Tables, page 7-101

Configuring a Destination Address Table

This task configures a dst-address routing table.

SUMMARY STEPS

1. configure terminal
2. sbc sbc-name
3. sbe
4. call-policy-set policy-set-id
5. first-call-routing-table table-name
6. rtg-dst-address-table table-name
7. entry entry-id
8. match-address key [regex | string | digits]
9. prefix
10. dst-adjacency target-adjacency
11. action [next-table goto-table-name | complete | reject]
12. exit
13. entry entry-id
14. match-address key [regex | string | digits]
15. prefix
16. dst-adjacency target-adjacency
17. action [next-table goto-table-name | complete | reject]
18. exit
19. entry entry-id
20. match-address key [regex | string | digits]
21. prefix
22. dst-adjacency target-adjacency
23. action [next-table goto-table-name | complete | reject]
24. exit
25. entry entry-id
26. match-address key [regex | string | digits]
27. prefix
28. dst-adjacency target-adjacency
29. action [next-table goto-table-name | complete | reject]
30. exit
31. complete name
32. end
33. show

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc sbc-name</td>
<td>Enters the mode of an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# sbc mysbc</td>
<td>• Use the <em>sbc-name</em> argument to define the name of the service.</td>
</tr>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Enters the mode of an SBE entity within an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> call-policy-set policy-set-id</td>
<td>Enters the mode of routing policy set configuration within an SBE entity.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe)# call-policy-set 1</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td><code>first-call-routing-table table-name</code></td>
<td>Configures the name of the first policy table to process when performing the routing stage of policy for new-call events.</td>
</tr>
<tr>
<td>6</td>
<td><code>rtg-dst-address-table table-name</code></td>
<td>Enters the configuration mode of a routing table within the context of an SBE policy set with the entries of the table matching the dialed number (after number analysis).</td>
</tr>
<tr>
<td>7</td>
<td><code>entry entry-id</code></td>
<td>Enters the mode for configuring an entry in a routing table, creating the entry, if necessary.</td>
</tr>
<tr>
<td>8</td>
<td>`match-address key</td>
<td>regex</td>
</tr>
<tr>
<td>9</td>
<td><code>prefix</code></td>
<td>Configures the match-address of this entry to match the start of the destination address.</td>
</tr>
<tr>
<td>10</td>
<td><code>dst-adjacency target-adjacency</code></td>
<td>Configures the destination adjacency of an entry in a routing table.</td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Step 11</th>
<th>action [next-table \ goto-table-name \ complete \ reject]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purpose</td>
<td>Configures the action to take if this routing entry is chosen. Possible actions are:</td>
</tr>
<tr>
<td></td>
<td>• Set the name of the next routing table to process if the event matches this entry. This is done using the <strong>next-table</strong> keyword and the <strong>goto-table-name</strong> argument.</td>
</tr>
<tr>
<td></td>
<td>• Complete the action using the <strong>complete</strong> keyword.</td>
</tr>
<tr>
<td></td>
<td>• Reject the indicated action using the <strong>reject</strong> keyword.</td>
</tr>
</tbody>
</table>

**Example:**

```
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)#
action complete
```

<table>
<thead>
<tr>
<th>Step 12</th>
<th>exit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purpose</td>
<td>Exits the <strong>entry</strong> mode to the <strong>rtgtable</strong> mode.</td>
</tr>
</tbody>
</table>

**Example:**

```
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)#
exit
```

<table>
<thead>
<tr>
<th>Step 13</th>
<th>entry entry-id</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purpose</td>
<td>Enters the mode for configuring an entry in a routing table, creating the entry, if necessary.</td>
</tr>
</tbody>
</table>

**Example:**

```
Router(config-sbc-sbe-rtgpolicy-rtgtable)#
entry 2
```

<table>
<thead>
<tr>
<th>Step 14</th>
<th>match-address key [regex \ string \ digits]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purpose</td>
<td>Configures the match value of an entry in a routing table.</td>
</tr>
<tr>
<td></td>
<td>To create a routing table that routes on user name, use the existing rtg-dst-address-table or rtg-src-address-table and put a textual value in the match-address field.</td>
</tr>
<tr>
<td></td>
<td>The SBC skips number analysis and performs only routing when the SIP message contains a user name. The SBC decides that an address is a user name (as opposed to a phone number) if it contains any character other than: 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, plus, hyphen, period, open-round-bracket, and close-round-bracket.</td>
</tr>
<tr>
<td></td>
<td>When the SBC has decided that an address is a user name, the “X” in the routing tables is treated not as a wildcard character, but as a literal “X”. For example, the match value of “X” matches the username “X”, but not “A”.</td>
</tr>
</tbody>
</table>

**Example:**

```
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)#
match-address 434
```

<table>
<thead>
<tr>
<th>Step 15</th>
<th>prefix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purpose</td>
<td>Configures the match-address of this entry to match the start of the destination address.</td>
</tr>
</tbody>
</table>

**Example:**

```
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)#
prefix
```

<table>
<thead>
<tr>
<th>Step 16</th>
<th>dst-adjacency target-adjacency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purpose</td>
<td>Configures the destination adjacency of an entry in a routing table.</td>
</tr>
</tbody>
</table>

**Example:**

```
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)#
dst-adjacency SIP-ASS40-PSTN-GW1
```
### How to Implement Policies

**Command or Action**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| 17   | action [next-table goto-table-name | complete | reject] | Configures the action to take if this routing entry is chosen. Possible actions are:  
- Set the name of the next routing table to process if the event matches this entry. This is done using the **next-table** keyword and the **goto-table-name** argument.  
- Complete the action using the **complete** keyword.  
- Reject the indicated action using the **reject** keyword. |
|      |          |         |
| 18   | exit     | Exits the entry mode to the rtgtable mode. |
|      |          |         |
| 19   | entry entry-id | Enters the mode for configuring an entry in a routing table, creating the entry, if necessary. |
|      |          |         |
| 20   | match-address key [regex | string | digits] | Configures the match value of an entry in a routing table.  
To create a routing table that routes on user name, use the existing rtg-dst-address-table or rtg-src-address-table and put a textual value in the match-address field.  
The SBC skips number analysis and performs only routing when the SIP message contains a user name. The SBC decides that an address is a user name (as opposed to a phone number) if it contains any character other than: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, plus, hyphen, period, open-round-bracket, and close-round-bracket.  
When the SBC has decided that an address is a user name, the “X” in the routing tables is treated not as a wildcard character, but as a literal “X”. For example, the match value of “X” matches the username “X”, but not “A”. |
<p>| | | |
|      |          |         |
| 21   | prefix   | Configures the match-address of this entry to match the start of the destination address. |
|      |          |         |
| 22   | dst-adjacency target-adjacency | Configures the destination adjacency of an entry in a routing table. |
|      |          |         |</p>
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 23</strong></td>
<td>Configures the action to take if this routing entry is chosen. Possible actions are:</td>
</tr>
</tbody>
</table>
| `action [next-table goto-table-name | complete | reject]` | - Set the name of the next routing table to process if the event matches this entry. This is done using the `next-table` keyword and the `goto-table-name` argument.  
  - Complete the action using the `complete` keyword.  
  - Reject the indicated action using the `reject` keyword. |
| Example: Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# action complete | |
| **Step 24** | Exits the `entry` mode to the `rtgtable` mode. |
| `exit` | |
| Example: Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# exit | |
| **Step 25** | Enters the mode for configuring an entry in a routing table, creating the entry, if necessary. |
| `entry entry-id` | |
| Example: Router(config-sbc-sbe-rtgpolicy-rtgtable)# entry 4 | |
| **Step 26** | Configures the match value of an entry in a routing table.  
To create a routing table that routes on user name, use the existing rtg-dst-address-table or rtg-src-address-table and put a textual value in the match-address field.  
The SBC skips number analysis and performs only routing when the SIP message contains a user name. The SBC decides that an address is a user name (as opposed to a phone number) if it contains any character other than: 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, plus, hyphen, period, open-round-bracket, and close-round-bracket.  
When the SBC has decided that an address is a user name, the “X” in the routing tables is treated not as a wildcard character, but as a literal “X”. For example, the match value of “X” matches the username “X”, but not “A”. |
| `match-address key [regex | string | digits]` | |
| Example: Router(config-sbc-sbe-rtgpolicy-rtgtable-rtgtable-entry)# match-address 454 | |
| **Step 27** | Configures the match-address of this entry to match the start of the destination address. |
| `prefix` | |
| Example: Router(config-sbc-sbe-rtgpolicy-rtgtable-rtgtable-entry)# prefix | |
| **Step 28** | Configures the destination adjacency of an entry in a routing table. |
| `dst-adjacency target-adjacency` | |
| Example: Router(config-sbc-sbe-rtgpolicy-rtgtable-rtgtable-entry)# dst-adjacency H323-AS540-PSTN-GW1 | |
### Command or Action

#### Step 29
```
action [next-table goto-table-name | complete | reject]
```
**Purpose**
Configure the action to take if this routing entry is chosen. Possible actions are:
- Set the name of the next routing table to process if the event matches this entry. This is done using the `next-table` keyword and the `goto-table-name` argument.
- Complete the action using the `complete` keyword.
- Reject the indicated action using the `reject` keyword.

#### Example:
```
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)#
action complete
```

#### Step 30
```
exit
```
**Purpose**
Exits the `entry` mode to the `rtgtable` mode.

#### Example:
```
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)#
exit
```

#### Step 31
```
complete name
```
**Purpose**
Completes the full routing policy set when you have committed the full set.

#### Example:
```
Router(config-sbc-sbe-rtgpolicy-rtgtable)#
complete
```

#### Step 32
```
end
```
**Purpose**
Exits `rtgtable` mode and enters Privileged Exec mode.

#### Example:
```
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)#
end
```

#### Step 33
```
show
```
**Purpose**
Displays the current configuration information.

#### Example:
```
Router# show
```

### Configuring the Destination, Source Domain, and Carrier ID Tables

This task configures dst-domain and src-domain and carrier ID routing tables.

### SUMMARY STEPS

1. `configure terminal`
2. `sbc sbc-name`
3. `sbe`
4. `call-policy-set policy-set-id`
5. `rtg-src-domain-table table-name` | `rtg-dst-domain-table table-name` | `rtg-carrier-id-table table-name`
6. `entry entry-id`
7. `match-domain key [regex]` | `match-cic cic`
8. `edit action`
9. edit-cic [del-prefix pd] | [del-suffix sd] | [add-prefix pa] | [replace ds]
10. action [next-table goto-table-name] | complete | reject
11. dst-adjacency target-adjacency
12. exit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc sbc-name</td>
<td>Enters the mode of an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# sbc mySBC</td>
<td>• Use the <em>sbc-name</em> argument to define the name of the service.</td>
</tr>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Enters the mode of an SBE entity within an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> call-policy-set policy-set-id</td>
<td>Enters the mode of routing policy set configuration within an SBE entity.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe)# call-policy-set 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> rtg-src-domain-table table-name</td>
<td>Enters the configuration mode of a routing table (creating a new table if necessary) whose entries match the source or destination domains, or carrier ID respectively.</td>
</tr>
<tr>
<td>rtg-dst-domain-table table-name</td>
<td>You are not allowed to enter the submode of routing table configuration in the context of the active policy set.</td>
</tr>
<tr>
<td>rtg-carrier-id-table table-name</td>
<td>The <em>no</em> version of the command destroys the routing table. A routing table may not be destroyed if it is in the context of the active policy set.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-rtgpolicy)# rtg-src-domain-table MyRtgTable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> entry entry-id</td>
<td>Enters the mode for configuring an entry in a routing table, creating the entry, if necessary.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-rtgpolicy-rtgtable)# entry 1</td>
<td><em>entry-id</em> is a number that uniquely identifies an entry in the newly created routing table.</td>
</tr>
<tr>
<td><strong>Step 7</strong> match-domain key [regex]</td>
<td>Creates or modifies the matching domain or carrier id code (CIC) of an entry in a routing table.</td>
</tr>
<tr>
<td>match-cic cic</td>
<td>• <em>key</em> is regular expression, not just a string.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# match-domain ^cisco.com$</td>
<td>• <em>cic</em> is the carrier ID that matches the entry in a routing table.</td>
</tr>
</tbody>
</table>
### Step 8

**edit action**

**Example:**

Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# edit del-prefix 1

- **Purpose:** Configures a dial-string manipulation action in the routing table. You are not allowed to do this if the table is part of the active policy set.

- **edit** command can be set to the following values:
  - `del-prefix pd` — Delete prefix `pd`, where `pd` is a positive integer specifying a number of digits to delete from the front of the dialed digit string.
  - `del-suffix sd` — Delete suffix `sd`, where `sd` is a positive integer specifying a number of digits to delete from the end of the dialed digit string.
  - `add-prefix pa` — Add prefix `pa`, where `pa` is a string of digits to add to the front of the dialed string.
  - `replace ds` — Replace `ds`, where `ds` is a string of digits that replaces the dialed string.

In the example to the left, the `edit` command sets entry 1 to delete 1 digit from the beginning of the dialed string in the routing table “MyRtgTable”.

### Step 9

**edit-cic [del-prefix pd] | [del-suffix sd] | [add-prefix pa] | [replace ds]**

**Example:**

Router(config-sbc-sbe-rtgpolicy-natable-entry)# edit-cic del-prefix 1

- **Purpose:** Configures a carrier identification code (CIC) manipulation action in any routing table.

- **Purpose:** You are not allowed to do this if the table is part of the active policy set.

- **del-prefix pd**: A positive integer specifying a number of digits to delete from the front of the carrier ID string.
- **del-suffix sd**: A positive integer specifying a number of digits to delete from the end of the carrier ID string.
- **add-prefix pa**: A string of digits to add to the front of the carrier ID string.
- **replace ds**: A string of digits to replace the carrier ID string with.

The following command sets entry 2 to delete the first digit of the carrier ID in the current routing table.

If you wish to remove the carrier ID entirely from outgoing messages, you should specify a replacement string of 0000 or a prefix deletion length of 4. For example,

```
edit-cic del-prefix 4   OR
edit-cic replace 0000
```
### Configuring the Category Table

This task configures dst-domain and src-domain and carrier ID routing tables.

#### SUMMARY STEPS

1. configure terminal
2. sbc sbc-name
3. sbe
4. call-policy-set policy-set-id
5. rtg-category-table table-name
6. entry entry-id
7. match-category word
8. action [next-table goto-table-name | complete | reject]
9. exit

#### Command or Action | Purpose
---|---
**Step 10** action [next-table goto-table-name | complete | reject] | Configures the action to take if this routing entry is chosen. Possible actions are:
- Set the name of the next routing table to process if the event matches this entry. This is done using the next-table keyword and the goto-table-name argument.
- Complete the action using the complete keyword.
- Reject the indicated action using the reject keyword.

**Example:**
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# action complete

**Step 11** dst-adjacency target-adjacency | Configures the destination adjacency of an entry in a routing table.

**Example:**
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# dst-adjacency SIP-AS540-PSTN-GW2

**Step 12** exit | Exits the current mode of the configuration.

**Example:**
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# exit
### How to Implement Policies

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc sbc-name</td>
<td>Enters the mode of an SBC service.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config)# sbc mysbc</td>
<td>Use the <code>sbc-name</code> argument to define the name of the service.</td>
</tr>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Enters the mode of an SBE entity within an SBC service.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> call-policy-set policy-set-id</td>
<td>Enters the mode of routing policy set configuration within an SBE entity.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc)# call-policy-set 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> rtg-category-table table-name</td>
<td>Enters the submode of configuration of a routing table whose entries match on the category within the context of an SBE policy set.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-rtgpolicy)# rtg-category-table MyRtgTable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> entry entry-id</td>
<td>Enters the mode for configuring an entry in a routing table, creating the entry, if necessary.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-rtgpolicy-rtgtable)# entry 1</td>
<td>entry-id is a number that uniquely identifies an entry in the newly created routing table.</td>
</tr>
<tr>
<td><strong>Step 7</strong> match-category word</td>
<td>Configures the match value of an entry in a routing table matching on the category.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# match-category emergency$</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> action [next-table goto-table-name</td>
<td>complete</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# action reject</td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong> exit</td>
<td>Exits the current mode of the configuration.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# exit</td>
<td></td>
</tr>
</tbody>
</table>
Configuring the Least Cost Table

This task configures a Least Cost routing table.

SUMMARY STEPS

1. `configure terminal`
2. `sbc sbc-name`
3. `sbe`
4. `call-policy-set policy-set-id`
5. `rtg-least-cost-table table-name`
6. `entry entry-id`
7. `cost cost`
8. `dst-adjacency`
9. `action complete`
10. `exit`

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure terminal</td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>Router# configure terminal</code></td>
</tr>
<tr>
<td>Step 2 sbc sbc-name</td>
<td>Enters the mode of an SBC service.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>Router(config)# sbc mysbc</code></td>
</tr>
<tr>
<td></td>
<td>• Use the <code>sbc-name</code> argument to define the name of the service.</td>
</tr>
<tr>
<td>Step 3 sbe</td>
<td>Enters the mode of an SBE entity within an SBC service.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>Router(config-sbc)# sbe</code></td>
</tr>
<tr>
<td>Step 4 call-policy-set policy-set-id</td>
<td>Enters the mode of routing policy set configuration within an SBE entity.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>Router(config-sbc-sbe)# call-policy-set 1</code></td>
</tr>
<tr>
<td>Step 5 rtg-least-cost-table table-name</td>
<td>Enters the submode of configuration of a routing table whose entries match on the least cost within the context of an SBE policy set.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>Router(config-sbc-sbe-rtgpolicy)# rtg-least-cost-table MyRtgTable</code></td>
</tr>
</tbody>
</table>
## How to Implement Policies

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 6</strong> entry entry-id</td>
<td>Enters the mode for configuring an entry in a routing table, creating the entry, if necessary. entry-id is a number that uniquely identifies an entry in the newly created routing table.</td>
</tr>
<tr>
<td><strong>Step 7</strong> cost cost</td>
<td>Assigns a cost to the route.</td>
</tr>
<tr>
<td><strong>Step 8</strong> dst-adjacency target-adjacency</td>
<td>Configures the destination adjacency of an entry in a routing table.</td>
</tr>
<tr>
<td><strong>Step 9</strong> action complete</td>
<td>Specifies that routing is complete when an entry matches this policy</td>
</tr>
<tr>
<td><strong>Step 10</strong> exit</td>
<td>Exits the current mode of the configuration.</td>
</tr>
</tbody>
</table>

Example:

- **Step 6**
  ```
  Router(config-sbc-sbe-rtgpolicy-rtgtable)# entry 1
  ```

- **Step 7**
  ```
  Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# cost 50
  ```

- **Step 8**
  ```
  Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# dst-adjacency SIP-AS540-PSTN-GW2
  ```

- **Step 9**
  ```
  Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# action complete
  ```

- **Step 10**
  ```
  Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# exit
  ```
Configuring Time-Based Tables

This task configures dst-domain and src-domain and carrier ID routing tables.

**SUMMARY STEPS**

1. `configure terminal`
2. `sbc sbc-name`
3. `sbe`
4. `call-policy-set policy-set-id`
5. `rtg-time-table table-name`
6. `entry entry-id`
7. `match-time {[date yr year_low year_high mon month_low month_high day date_low date_high] [dow DoW_low DoW_high] [tod hr hour_low hour_high min minute_low minute_high]}`
8. `precedence precedence`
9. `dst-adjacency dst_adj`
10. `action [next-table goto-table-name | complete | reject]`
11. `exit`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>configure terminal</code></td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>sbc sbc-name</code></td>
<td>Enters the mode of an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# sbc mysbc</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>sbe</code></td>
<td>Enters the mode of an SBE entity within an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> <code>call-policy-set policy-set-id</code></td>
<td>Enters the mode of routing policy set configuration within an SBE entity.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe)# call-policy-set 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> <code>rtg-time-table table-name</code></td>
<td>Enters the submode of configuration of a routing table whose entries match on the time within the context of an SBE policy set.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-rtgpolicy)# rtg-time-table MyRtgTable</td>
<td></td>
</tr>
</tbody>
</table>
How to Implement Policies

**Step 6**

```
entry entry-id
```

**Example:**

```
Router(config-sbc-sbe-rtgpolicy-rtgtable)# entry 1
```

Enters the mode for configuring an entry in a routing table, creating the entry, if necessary.  
*entry-id* is a number that uniquely identifies an entry in the newly created routing table.

**Step 7**

```
match-time {[/date yr year_low year_high mon month_low month_high day date_low date_high] [dow DoW_low DoW_high] [tod hr hour_low hour_high min minute_low minute_high]}
```

**Example:**

```
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# match-time date yr 2006 2020 mon 1 12 day 1 31
```

Configures the match time of an entry. A string used to match the time and can include one or more of the following specifiers:

- *date_low - date_high*: the inclusive range of dates (1-31).
- *date*: date
- *day*: date
- *DoW_low - DoW_high*: the inclusive range of days (Sun-Mon).
- *dow*: day of the week
- *hr*: hour
- *hour_low - hour_high*: the inclusive range of hours (0-23).
- *minute_low - minute_high*: the inclusive range of minutes (0-59).
- *min*: minute
- *mon*: month
- *month_low - month_high*: the inclusive range of months (1-12).
- *tod*: time of day
- *yr*: year
- *year_low - year_high*: the inclusive range of years.

The high values are optional and if unspecified are set equal to the low values.

**Step 8**

```
predence precedence
```

**Example:**

```
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# precedence 0
```

Configures the precedence of the routing entry.

**Step 9**

```
action [next-table goto-table-name | complete | reject]
```

**Example:**

```
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# action complete
```

Configures the action to take if this routing entry is chosen. Possible actions are:

- Set the name of the next routing table to process if the event matches this entry. This is done using the *next-table* keyword and the *goto-table-name* argument.
- Complete the action using the *complete* keyword.
- Reject the indicated action using the *reject* keyword.
Chapter 7  Implementing Cisco Unified Border Element (SP Edition) Policies

How to Implement Policies

Configuring Trunk-Group ID Tables

This task configures src-trunk-group-id and dst-trunk-group-id routing tables.

SUMMARY STEPS

1. configure terminal
2. sbc sbc-name
3. sbe
4. adjacency sip adjacency-name
5. tgid-routing
6. exit
7. call-policy-set policy-set-id
8. rtg-src-trunk-group-id-table table-name | rtg-dst-trunk-group-id-table table-name
9. entry entry-id
10. action {next-table goto-table-name | complete | reject}
11. dst-adjacency dst_adj
12. match-type {none | any | context | tgid}
13. tgid-context tgid-context-name {tgid tgid-name}
14. exit
## How to Implement Policies

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>configure terminal</strong></td>
</tr>
</tbody>
</table>
| **Example:**      | Router# configure terminal  
|                   | Router(config)# |
| **Step 2**        | **sbc sbc-name** |
| **Example:**      | Router(config)# sbc mysbc |
| **Step 3**        | **sbe** |
| **Example:**      | Router(config-sbc)# sbe  
|                   | Router(config-sbc-sbe)# |
| **Step 4**        | **adjacency sip adjacency-name** |
| **Example:**      | Router(config-sbc-sbe)# adjacency sip adj1  
|                   | Router(config-sbc-sbe-adj-sip)# |
| **Step 5**        | **tgid-routing** |
| **Example:**      | Router(config-sbc-sbe-adj-sip)# tgid-routing  
|                   | Router(config-sbc-sbe-adj-sip)# |
| **Step 6**        | **exit** |
| **Example:**      | Router(config-sbc-sbe-adj-sip)# exit  
|                   | Router(config-sbc-sbe)# |
| **Step 7**        | **call-policy-set policy-set-id** |
| **Example:**      | Router(config-sbc-sbe)# call-policy-set 1  
|                   | Router(config-sbc-sbe-rtgpolicy)# |
| **Step 8**        | **rtg-src-trunk-group-id-table table-name** |
| **Example:**      | Router(config-sbc-sbe-rtgpolicy)#  
|                   | rtg-src-trunk-group-id-table MyRtgTable  
|                   | Router(config-sbc-sbe-rtgpolicy-rtgtable)# |
### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>entry entry-id</td>
<td>Enters the mode for configuring an entry in a routing table, creating the entry, if necessary. <em>entry-id</em> is a number that uniquely identifies an entry in the newly created routing table.</td>
</tr>
<tr>
<td>10</td>
<td>action [next-table goto-table-name</td>
<td>Configures the action to take if this routing entry is chosen. Possible actions are:</td>
</tr>
<tr>
<td></td>
<td>complete</td>
<td></td>
</tr>
<tr>
<td></td>
<td>reject]</td>
<td>• Set the name of the next routing table to process if the event matches this entry. This is done using the <em>next-table</em> keyword and the <em>goto-table-name</em> argument.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Complete the action using the <em>complete</em> keyword.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Reject the indicated action using the <em>reject</em> keyword.</td>
</tr>
<tr>
<td>11</td>
<td>dst-adjacency dst_adj</td>
<td>Configures the destination adjacency of an entry in a routing table.</td>
</tr>
<tr>
<td>12</td>
<td>match-type {none</td>
<td>any</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <em>none</em>: Matches an entry if no TGID information is present.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <em>any</em>: Matches an entry if any TGID information is present.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <em>context</em>: Matches an entry on the TGID context.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <em>tgid</em>: Matches an entry on both the TGID and TGID context.</td>
</tr>
<tr>
<td>13</td>
<td>tgid-context tgid-context-name {tgid tgid-name}</td>
<td>Defines trunk-group ID context and trunk-group ID to match the entries of the routing table.</td>
</tr>
<tr>
<td>14</td>
<td>exit</td>
<td>Exits the current mode of the configuration.</td>
</tr>
</tbody>
</table>
Configuring Number Manipulation

This task enables you to specify various number manipulations that can be performed on a dialed number after a destination adjacency has been selected.

SUMMARY STEPS

1. configure terminal
2. sbc sbc-name
3. sbe
4. call-policy-set policy-set-id
5. rtg-src-address-table table-id
6. rtg-src-adjacency-table table-id
7. rtg-src-account-table table-id
8. rtg-round-robin-table table-id
9. rtg-carrier-id-table table-id
10. rtg-dst-address-table table-id
11. entry entry-id
12. edit action
13. edit-cic [del-prefix pd] | [del-suffix sd] | [add-prefix pa] | [replace ds]
14. edit-src [del-prefix pd] | [del-suffix sd] | [add-prefix pa] | [replace ds]
15. exit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure terminal</td>
<td>Enables the global configuration mode.</td>
</tr>
</tbody>
</table>

**Example:**
Router# configure terminal

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 2 sbc sbc-name</td>
<td>Enters the mode of an SBC service.</td>
</tr>
</tbody>
</table>

**Example:**
Router(config)# sbc mysbc

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 3 sbe</td>
<td>Enters the mode of an SBE entity within an SBC service.</td>
</tr>
</tbody>
</table>

**Example:**
Router(config-sbc)# sbe

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 4 call-policy-set policy-set-id</td>
<td>Enters the mode of the routing policy set configuration in the SBE mode, creating a new policy set if necessary</td>
</tr>
</tbody>
</table>

**Example:**
Router(config-sbc-sbe)# call-policy-set 1
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 5</strong></td>
<td><strong>rtg-src-address-table</strong> table-id</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>Enters the configuration mode of a routing table (creating one if necessary) whose entries match the dialer’s number or SIP user name within the context of an SBE policy set.</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>You are not allowed to enter the submode of routing table configuration in the context of the active policy set.</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>The no version of the command destroys the routing table.</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>A routing table may not be destroyed if it is in the context of the active policy set.</strong></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td><strong>rtg-src-adjacency-table</strong> table-id</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>Enters the configuration mode of a routing table (creating one if necessary) within the context of an SBE policy set whose entries match the source adjacency.</strong></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td><strong>rtg-src-account-table</strong> table-id</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>Enters the configuration mode of a routing table (creating one if necessary) whose entries match the source account within the context of an SBE policy set.</strong></td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td><strong>rtg-round-robin-table</strong> table-id</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>Enters the configuration mode of a policy table, whose events do not have any match-value parameters, nor next-table actions. Its actions are restricted to configuring number manipulation, as well as setting the destination adjacency. A group of adjacencies are chosen for an event if an entry in a routing table matches that event and points to a round-robin adjacency table in the next-table action.</strong></td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td><strong>rtg-carrier-id-table</strong> table-id</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>Enters the configuration mode of a routing table (creating one if necessary) within the context of an SBE policy set whose entries match the carrier ID.</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>You are not allowed to enter the mode of the routing table configuration in the context of the active policy set.</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>The no version of the command destroys the routing table.</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>A routing table may not be destroyed if it is in the context of the active policy set.</strong></td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td><strong>rtg-dst-address-table</strong> table-id</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>Enters the configuration mode of a routing table (creating one if necessary) within the context of an SBE policy set whose entries match the dialed number (after number analysis) or SIP user name.</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>You are not allowed to enter the submode of routing table configuration in the context of the active policy set.</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>The no version of the command destroys the routing table.</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>A routing table may not be destroyed if it is in the context of the active policy set.</strong></td>
</tr>
</tbody>
</table>
## How to Implement Policies

### Step 11
**entry entry-id**

**Example:**
```bash
Router(config-sbc-sbe-rtgpolicy-rtgtable)# entry 1
```

Enters the mode for configuring an entry in a routing table, creating the entry if necessary.

### Step 12
**edit action**

**Example:**
```bash
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# edit del-prefix 1
```

Configures a dial-string manipulation action in the routing table. You are not allowed to do this if the table is part of the active policy set.

The **no** version of the command deletes the edit action of the given entry in the routing table.

The **edit** command can be set to the following values:

- **del-prefix pd** — Delete prefix `pd`, where `pd` is a positive integer specifying a number of digits to delete from the front of the dialed digit string.
- **del-suffix sd** — Delete suffix `sd`, where `sd` is a positive integer specifying a number of digits to delete from the end of the dialed digit string.
- **add-prefix pa** — Add prefix `pa`, where `pa` is a string of digits to add to the front of the dialed string.
- **replace ds** — Replace `ds`, where `ds` is a string of digits that replaces the dialed string.

In the example to the left, the **edit** command sets entry 1 to delete 1 digit from the beginning of the dialed string in the routing table “MyRtgTable”.

### Step 13
**edit-cic [del-prefix pd] | [del-suffix sd] | [add-prefix pa] | [replace ds]**

**Example:**
```bash
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# edit-cic del-prefix 1
```

Configures a CIC manipulation action in any routing table. You are not allowed to do this if the table is part of the active policy set.

- **del-prefix pd**: A positive integer specifying a number of digits to delete from the front of the carrier ID string.
- **del-suffix sd**: A positive integer specifying a number of digits to delete from the end of the carrier ID string.
- **add-prefix pa**: A string of digits to add to the front of the carrier ID string.
- **replace ds**: A string of digits to replace the carrier ID string with.

The following command sets entry 2 to delete the first digit of the carrier ID in the current routing table.

If you wish to remove the carrier ID entirely from outgoing messages, you should specify a replacement string of 0000 or a prefix deletion length of 4. For example,
```bash
edit-cic del-prefix 4   OR
edit-cic replace 0000
```
Chapter 7  Implementing Cisco Unified Border Element (SP Edition) Policies

How to Implement Policies

**Configuring Hunting**

This task enables Cisco Unified Border Element (SP Edition) to hunt for other routes or destination adjacencies in case of a failure.

**SUMMARY STEPS**

1. configure terminal
2. sbc sbc-name
3. sbe
4. adjacency sip adjacency-name or adjacency h323 adjacency-name
5. hunting-trigger error-codes or hunting-trigger error-codes
6. exit
7. h323
8. hunting-mode [altEndps | multiARQ]
9. end
10. show sbc sbc-name sbe h323 | sip hunting-trigger
11. show sbc sbc-name sbe h323 | sip hunting-mode

**Command or Action**

<table>
<thead>
<tr>
<th>Step 14</th>
<th>edit-src [del-prefix pd]</th>
<th>del-suffix sd</th>
<th>add-prefix pa</th>
<th>replace ds</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-rtgpolicy-natable-entry)# edit-src del-prefix 1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Purpose**

- Configures a source number manipulation action in the routing table.
- You are not allowed to do this if the table is part of the active policy set.
- The no version of the command destroys the match value.
  - del-prefix pd: A positive integer specifying a number of digits to delete from the front of the carrier ID string.
  - del-suffix sd: A positive integer specifying a number of digits to delete from the end of the carrier ID string.
  - add-prefix pa: A string of digits to add to the front of the carrier ID string.
  - replace ds: A string of digits to replace the carrier ID string with.

<table>
<thead>
<tr>
<th>Step 15</th>
<th>exit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-rtgpolicy-rtgtable-entry) # exit</td>
</tr>
</tbody>
</table>

**Exit**

- Exits the entry mode of the configuration.
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td><code>configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router# configure terminal</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Enters the mode of an SBC service.</td>
</tr>
<tr>
<td><code>sbc sbc-name</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config)# sbc mysbc</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Enters the mode of an SBE entity within an SBC service.</td>
</tr>
<tr>
<td><code>sbe</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc)# sbe</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Configures a destination SIP or H.323 adjacency for the SBC service, and enters into adjacency sip or adjacency h323 configuration mode.</td>
</tr>
<tr>
<td><code>adjacency sip adjacency-name</code> or <code>adjacency h323 adjacency-name</code></td>
<td></td>
</tr>
</tbody>
</table>
| **Example:** | Router(config-sbc-sbe)# adjacency sip test  
Router(config-sbc-sbe)# adjacency h323 test |

*adjacency sip*—A destination SIP adjacency where the configured failure return codes cause hunting to occur. This command overrides any globally configured retry error codes.

*adjacency h323*—A destination H.323 adjacency where the configured failure return codes cause hunting to occur. This command overrides any globally configured retry error codes.
### Command or Action

**Step 5**

- `hunting-trigger error-codes`
- `or`
- `hunting-trigger error-codes`

**Example:**

Router(config-sbc-sbe-adj-sip)# hunting-trigger 415 480

(This command configures the hunting trigger for a SIP adjacency in Adjacency SIP configuration mode.)

or

Router(config-sbc-sbe-adj-h323)#

hunting-trigger noBandwidth

Router(config-sbc-sbe-adj-h323)#

hunting-trigger unreachableDestination

(These commands configure the hunting trigger for an H.323 adjacency in Adjacency H.323 configuration mode.)

**Note**

If both adjacency level and SBE level hunting triggers are configured, the adjacency level takes priority.

### Purpose

Configures which failure return codes cause hunting to occur, in one of the following four modes:

- **sip** (global SIP scope)—use the `sip hunting-trigger` command.

**Note**

Exit (config-sbc-sbe-adj-sip) or (config-sbc-sbe-adj-h323) mode first and enter into (config-sbc-sbe) mode to configure in the global SIP scope level.

- **h323** (global H.323 scope)—use the `hunting-trigger` command
- **adjacency sip** (destination SIP adjacency)—use the `hunting-trigger` command
- **adjacency h323** (destination H.323 adjacency)—use the `hunting-trigger` command

**error-codes** can have the following values:

- **In the sip and adjacency sip modes**, `error-codes` represent a space-separated list of SIP numeric error codes. The examples to the left configures SIP to retry routing if it receives a “415” (media unsupported) or “480” (temporarily unavailable) error. Both error codes are set as hunting triggers. See Table 7-2 on page 7-22 for a list of SIP error codes.

- **In the h323 and adjacency h323 modes**, `error-codes` are entered in separate commands. The following is a list of H.323 textual error codes:
  - noBandwidth—The bandwidth is taken away or the ARQ is denied.
  - unreachableDestination—The terminal cannot reach the gatekeeper for ARQ.
  - destinationRejection—The code has been rejected at destination.
  - noPermission—The callee’s gatekeeper rejects the code.
  - gatewayResources—The gateway resources are exhausted.
  - badFormatAddress—The address field in the H.225 message is not understood.
  - securityDenied—The security settings are incompatible.
### Command or Action

#### Purpose

- the internally-defined value
  “connectFailed”—Either a releaseComplete response was received that gave no cause or any reason code for the release, or there was no response from the remote endpoint.

**Note** These textual error codes apply to H.323 only.

If you type `no sip hunting-trigger` or `no hunting-trigger`, then all error codes are cleared out. If you type `no sip hunting-trigger x y`, then just the codes x and y are removed from the configured list.

**Note** In the case of the `adjacency h323` mode, enter the `noRetry` value to specify that routing should never be retried for this adjacency no matter what failure return code is received.

### Step 6

**exit**

**Example:**

```
Router(config-sbc-sbe-adj-h323)# exit
```

Exits the Adjacency H.323 configuration mode and enters into SBE configuration mode.

### Step 7

**h323**

**Example:**

```
Router(config-sbc-sbe)# h323
```

The `h323` command enters into the H.323 configuration mode.

### Step 8

**hunting-mode [altEndps/multiARQ]**

**Example:**

```
Router(config-sbc-sbe-h323)# hunting-mode multiARQ
```

Configures the form of H.323 hunting to perform if H.323 hunting is triggered.

- **altEndps**—alternateEndpoints
- **multiARQ**—uses a nonstandard H.323 mechanism based on issuing multiple ARQs to a Gatekeeper for a single call.

The `no` version of this command restores the hunting mode to the default of alternateEndpoints. It does not disable hunting completely. If the hunting mode is not defined, the default is alternateEndpoints.

### Step 9

**end**

**Example:**

```
Router(config-sbc-sbe-h323)# end
```

Exits the current mode of the configuration and enters into Privileged EXEC mode.
### Activating a Routing Policy Set

This task activates a number analysis and routing policy set.

#### SUMMARY STEPS

1. `configure terminal`
2. `sbc sbc-name`
3. `sbe`
4. `call-policy-set default policy-set-id [priority priority-value]`

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>Purpose</strong></td>
</tr>
<tr>
<td><code>configure terminal</code></td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router# configure terminal</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><strong>Purpose</strong></td>
</tr>
<tr>
<td><code>sbc sbc-name</code></td>
<td>Enters the mode of an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config)# sbc mysbc</td>
</tr>
</tbody>
</table>

**Example:**

Router# show sbc mysbc sbe h323 hunting-trigger

Shows the H.323 or SIP hunting triggers.

**Example:**

Router# show sbc mysbc sbe h323 hunting-mode

Shows the H.323 hunting mode.
How to Implement Policies

Configuring H.323 MultiARQ Hunting

This task configures Cisco Unified Border Element (SP Edition) to hunt for other H.323 routes or destination adjacencies in case of a failure.

SUMMARY STEPS

1. configure terminal
2. sbc sbc-name
3. sbe
4. adjacency h323 adjacency-name
5. hunting-trigger error-codes
6. hunting-mode mode
7. exit
8. show sbc sbc-name sbe h323 hunting-mode
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc sbc-name</td>
<td>Enters the mode of an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# sbc mysbc</td>
<td>• Use the <code>sbc-name</code> argument to define the name of the service.</td>
</tr>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Enters the mode of an SBE entity within an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> adjacency h323 adjacency-name</td>
<td>Configures a destination H.323 adjacency for the SBC service, and enters into adjacency h323 configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe)# adjacency h323 test</td>
<td>A destination H.323 adjacency is where the configured failure return codes cause hunting to occur. This command overrides any globally configured retry error codes.</td>
</tr>
</tbody>
</table>
## How to Implement Policies

### Step 5

**Command or Action**

`hunting-trigger error-codes`

**Example:**

```bash
Router(config-sbc-sbe-h323)# hunting-trigger
noBandwidth
Router(config-sbc-sbe-h323)# hunting-trigger
securityDenied
```

**Purpose**

Configures which failure return codes cause hunting to occur, in one of the following configuration modes:

- `h323` (global H.323 scope)
- `adjacency h323` (destination H.323 adjacency)

The example to the left configures H.323 to retry routing if it receives a “noBandwidth” or “securityDenied” error codes.

In the `h323` and `adjacency h323` configuration modes, `error-codes` are entered in separate commands. The following is a list of H.323 textual error codes:

- noBandwidth
- unreachableDestination
- destinationRejection
- noPermission
- gatewayResources
- badFormatAddress
- securityDenied
- the internally-defined value “connectFailed”

If you type `no hunting-trigger`, all error codes are cleared out.

**Note**

In the case of the `adjacency h323` mode, enter the `noRetry` value to specify that routing should never be retried for this adjacency no matter what failure return code is received.

### Step 6

**Command or Action**

`hunting-mode [altEndps|multiARQ]`

**Example:**

```bash
Router(config-sbc-sbe-h323)# hunting-mode
multiARQ
```

**Purpose**

Configures the form of hunting to perform if hunting is triggered.

- `altEndps`—alternateEndpoints
- `multiARQ`—uses a nonstandard H.323 mechanism based on issuing multiple ARQs to a Gatekeeper for a single call.

The `no` version of this command restores the hunting mode to the default of alternateEndpoints. It does not disable hunting completely. If the hunting mode is not defined, the default is alternateEndpoints.

### Step 7

**Command or Action**

`exit`

**Example:**

```bash
Router(config-sbc-sbe-h323)# exit
```

**Purpose**

Exits the current mode of the configuration and enters into Privileged EXEC mode.

### Step 8

**Command or Action**

`show sbc sbc-name sbe h323 hunting-mode`

**Example:**

```bash
Router# show sbc mysbc sbe h323 hunting-mode
```

**Purpose**

Shows the H.323 hunting mode.
Configuring Call Admission Control Policy Sets, CAC Tables, and Global CAC Policy Sets

This optional task configures Call Admission Control policy sets, CAC tables, and assigns a global CAC policy set.

**Note**
If you are performing this procedure to modify an active CAC policy set, see the [Modifying Active CAC Policy Sets](#) section on page 7-8 prior to performing the procedure.

**SUMMARY STEPS**

1. configure terminal
2. sbc sbc-name
3. sbe
4. cac-policy-set averaging-period avg-number avg-period
5. cac-policy-set policy-set-id
6. first-cac-scope scope-name
7. first-cac-table table-name
8. cac-table table-name
9. table-type {policy-set | limit {list of limit tables}}
10. entry entry-id
11. cac-scope {list of scope options}
12. match-value key
13. max-num-calls mnc
14. max-call-rate-per-scope limit [averaging-period period-num]
15. max-in-call-msg-rate limit [averaging-period period-num]
16. max-out-call-msg-rate limit [averaging-period period-num]
17. max-bandwidth mbw bwsize
18. callee-privacy callee-priv-setting
19. action [next-table goto-table-name | cac-complete]
20. exit
21. entry entry-id
22. match-value key
23. max-num-calls mnc
24. max-call-rate-per-scope limit [averaging-period period-num]
25. max-bandwidth mbw bwsize
26. transcode-deny
27. max-regs-rate-per-scope limit [averaging-period period-num]
28. action [next-table goto-table-name | cac-complete]
29. `exit`
30. `exit`
31. `complete`
32. `exit`
33. `cac-policy-set global cac-policy-num`
34. `end`
35. `show sbc sbc-name sbe cac-policy-set [global]`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc sbc-name</td>
<td>Enters the mode of an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# sbc mysbc</td>
<td>• Use the <code>sbc-name</code> argument to define the name of the service.</td>
</tr>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Enters the mode of an SBE entity within an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> cac-policy-set averaging-period avg-number avg-period</td>
<td>Specifies the averaging period for rate calculations.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe)# cac-policy-set averaging-period 1 100</td>
<td>• <code>avg-number</code>—The averaging period number, can be 1 or 2.</td>
</tr>
<tr>
<td>Router(config-sbc-sbe)# cac-policy-set averaging-period 2 175</td>
<td>• <code>avg-period</code>—The averaging period used by CAC in rate calculations in seconds, can range from 1 to 3600 seconds. By default, 60 seconds is configured.</td>
</tr>
<tr>
<td><strong>Step 5</strong> cac-policy-set policy-set-id</td>
<td>Enters the mode of CAC policy set configuration within an SBE entity, creating a new policy set if necessary.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe)# cac-policy-set 1</td>
<td>• <code>policy-set-id</code>—The call policy set number that can range from 1 to 2147483647.</td>
</tr>
</tbody>
</table>
**Step 6**  
**first-cac-scope**  
*Scope-name*  

**Example:**  
Router(config-sbc-sbe-cacpolicy)#  
first-cac-scope global

- Configures the scope at which to begin defining limits when performing the admission control stage of policy.  
- **Note:** The first-cac-scope definition is only relevant if the table type configured by the first-cac-table command is a Limit table. In that case, the scope of the first-cac-table is determined by first-cac-scope. If the first-cac-table is a Policy Set table, the first-cac-scope is ignored and defaults to global.

The *scope-name* argument configures the scope at which limits should be initially defined. Possible values are:

- adj-group  
- call  
- category  
- dst-account  
- dst-adj-group  
- dst-adjacency  
- dst-number  
- global  
- src-account  
- src-adj-group  
- src-adjacency  
- src-number

Features can be enabled or disabled per adjacency group through CAC configuration the same way this is done per individual adjacencies.

**Step 7**  
**first-cac-table**  
*Table-name*

**Example:**  
Router(config-sbc-sbe-cacpolicy)#  
first-cac-table StandardListByAccount

- Configures the name of the first policy table to process when performing the admission control stage of policy.

**Step 8**  
**cac-table**  
*Table-name*

**Example:**  
Router(config-sbc-sbe-cacpolicy)#  
cac-table StandardListByAccount

- Enters the mode for configuration of an admission control table (creating one if necessary) within the context of an SBE policy set.
### How to Implement Policies

**Step 9**  
**table-type** *(policy-set | limit) *(list of limit tables))

**Example:**  
Router(config-sbc-sbe-cacpolicy-cactable)#  
**table-type** policy-set

---

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>table-type</strong> *(policy-set</td>
<td>limit) *(list of limit tables))</td>
</tr>
<tr>
<td></td>
<td>• account—Compare the name of the account.</td>
</tr>
<tr>
<td></td>
<td>• adj-group—Compare the name of the adjacency group.</td>
</tr>
<tr>
<td></td>
<td>• adjacency—Compare the name of the adjacency.</td>
</tr>
<tr>
<td></td>
<td>• all—No comparison type. All events match this type.</td>
</tr>
<tr>
<td></td>
<td>• call-priority—Compare with call priority.</td>
</tr>
<tr>
<td></td>
<td>• category—Compare the number analysis assigned category.</td>
</tr>
<tr>
<td></td>
<td>• dst-account—Compare the name of the destination account.</td>
</tr>
<tr>
<td></td>
<td>• dst-adj-group—Compare the name of the destination adjacency group.</td>
</tr>
<tr>
<td></td>
<td>• dst-adjacency—Compare the name of the destination adjacency.</td>
</tr>
<tr>
<td></td>
<td>• dst-prefix—Compare the beginning of the dialed digit string.</td>
</tr>
<tr>
<td></td>
<td>• event-type—Compare with CAC policy event types.</td>
</tr>
<tr>
<td></td>
<td>• src-account—Compare the name of the source account.</td>
</tr>
<tr>
<td></td>
<td>• src-adj-group—Compare the name of the source adjacency group.</td>
</tr>
<tr>
<td></td>
<td>• src-adjacency—Compare the name of the source adjacency.</td>
</tr>
<tr>
<td></td>
<td>• src-prefix—Compare the beginning of the calling number string.</td>
</tr>
</tbody>
</table>

**Note**  
For Limit tables, the event or message or call matches only a single entry.

Features can be enabled or disabled per adjacency group through CAC configuration the same way this is done per individual adjacencies. The adj-group table type matches on either source or destination adjacency group.

When the policy-set keyword is specified, use the **cac-scope** command to configure the scope within each entry at which limits are applied in a CAC Policy Set table.

**Note**  
For Policy Set tables, the event or call or message is applied to all entries in this table.
### How to Implement Policies

<table>
<thead>
<tr>
<th>Step 10</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><code>entry entry-id</code></td>
<td>Enters the mode to create or modify an entry in an admission control table.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-cacpolicy-cactable)# entry 1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 11</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><code>cac-scope {list of scope options}</code></td>
<td>Configures the scope within each of the entries at which limits are applied in a policy set table.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-cacpolicy-cactable-entry)# cac-scope category</td>
<td></td>
</tr>
</tbody>
</table>

- **list of scope options**—Specifies one of the following strings used to match events:
  - `account`—Events that are from the same account.
  - `adjacency`—Events that are from the same adjacency.
  - `adj-group`—Events that are from members of the same adjacency group.
  - `call`—Scope limits are per single call.
  - `category`—Events that have same category.
  - `dst-account`—Events that are sent to the same account.
  - `dst-adj-group`—Events that are sent to the same adjacency group.
  - `dst-adjacency`—Events that are sent to the same adjacency.
  - `dst-number`—Events that have same destination.
  - `global`—Scope limits are global
  - `src-account`—Events that are from the same account.
  - `src-adj-group`—Events that are from the same adjacency group.
  - `src-adjacency`—Events that are from the same adjacency.
  - `src-number`—Events that have the same source number.
  - `sub-category`—The limits specified in this scope apply to all events sent to or received from members of the same subscriber category.
  - `sub-category-pfx`—The limits specified in this scope apply to all events sent to or received from members of the same subscriber category prefix.
  - `subscriber`—The limits specified in this scope apply to all events sent to or received from individual subscribers (a device that is registered with a Registrar server).
## How to Implement Policies

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 12** match-value key | Configures the match-value of an entry in a CAC Limit table. It is only relevant for Limit table types. The key argument is a string or a keyword based on the table type. The format of the key is determined by the Limit table type (for example, Limit event-type tables or Limit call-priority tables). For Limit event-type tables *(table-type limit event-type)*, the match value string options are the following:  
  - *call-update*—Compare the beginning of the calling number string.  
  - *endpoint-reg*—Compare the name of the destination adjacency.  
  - *new-call*—Compare the beginning of the dialed digit string.  
  
For Limit call-priority tables *(table-type limit call-priority)*, the match value string options are the following:  
  - *critical*—Match calls with resource priority 'critical'.  
  - *flash*—Match calls with resource priority 'flash'.  
  - *flash-override*—Match calls with resource priority 'flash-override'.  
  - *immediate*—Match calls with resource priority 'immediate'.  
  - *priority*—Match calls with resource priority 'priority'.  
  - *routine*—Match calls with resource priority 'routine'.  

For all other Limit tables, enter a name or digit string:  
  - *WORD*—Name or digit string to match. (Max Size 255). |
| **Example:** Router(config-sbc-sbe-cacpolicy-cactable-entry)# match-value SIP-CUSTOMER-1 |  |

| **Step 13** max-num-calls mnc | Configures the maximum number of calls of an entry in an admission control table. |
| **Example:** Router(config-sbc-sbe-cacpolicy-cactable-entry)# max-num-calls 100 |  |

| **Step 14** max-call-rate-per-scope limit [averaging-period period-num] | Configures the maximum call rate for an entry in an admission control table.  
  - *limit*—The limit for the number of new calls per minute. The value can range from 0 to 2147483647.  
  - *averaging-period*—Specifies the averaging-period to use in the rate calculation. By default, 1 is selected.  
  - *period-num*—Calculates rate based on specified averaging period, ranging from 1 to 2. |
| **Example:** Router(config-sbc-sbe-cacpolicy-cactable-entry)# max-call-rate-per-scope 1000 averaging-period 2 |  |
Chapter 7  Implementing Cisco Unified Border Element (SP Edition) Policies

How to Implement Policies

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 15**

`max-in-call-msg-rate limit [averaging-period period-num]`

**Example:**
Router(config-sbc-sbe-cacpolicy-cactable-entry)
# max-in-call-msg-rate 500 averaging-period 2

Confirms the maximum in call rate for an entry in an admission control table.
- **limit**—The limit for the number of in-call messages per minute. The value can range from 0 to 2147483647.
- **averaging-period**—Specifies the averaging-period to use in the rate calculation. By default, 1 is selected.
- **period-num**—Calculates rate-based on specified averaging period, ranging from 1 to 2.

| **Step 16**

`max-out-call-msg-rate limit [averaging-period period-num]`

**Example:**
Router(config-sbc-sbe-cacpolicy-cactable-entry)
# max-out-call-msg-rate 500 averaging-period 2

Confirms the maximum out call rate for an entry in an admission control table.
- **limit**—The limit for the number of new calls per minute. The value can range from 0 to 2147483647.
- **averaging-period**—Specifies the averaging-period to use in the rate calculation. By default, 1 is selected.
- **period-num**—Calculates rate-based on specified averaging period, ranging from 1 to 2.

| **Step 17**

`max-bandwidth mbw bwsize`

**Example:**
Router(config-sbc-sbe-cacpolicy-cactable-entry)
# max-bandwidth 1000000 bps

Confirms the maximum bidirectional bandwidth for an entry in an admission control table. For example, if a max-bandwidth value is configured, the SBC allows half of this value in each direction.

The `mbw` argument is a positive integer specifying the total maximum rate at which call media should be admitted in both directions (in bytes per second).

The `bwsize` argument specifies the transfer size to which `mbw` refers. Possible values are:
- bps
- Kbps
- Mbps
- Gbps

| **Step 18**

`callee-privacy [callee-priv-setting]`

**Example:**
Router(config-sbc-sbe-cacpolicy-cactable-entry)
# callee-privacy never

Confirms the level of privacy processing to perform on messages sent from callee to caller.

The `callee_priv_setting` argument indicates the specific callee privacy setting. Possible values are:
- never—Indicates to never hide identity.
- account-boundary—Indicates to hide identity only if caller is different account from callee.
- always—Indicates to always hide identity.
### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| 19   | **action [next-table goto-table-name | cac-complete]** | Configures the action to perform after this entry in an admission control table. Possible actions are:  
- Identify the next CAC table to process using the `next-table` keyword and the `goto-table-name` argument.  
- Stop processing for this scope using the `cac-complete` keyword. |
|      | **Example:**     |         |
|      | Router(config-sbc-sbe-cacpolicy-cactable-entry) # action cac-complete |         |
| 20   | **exit**         | Exits from `entry` to `cactable` mode. |
|      | **Example:**     |         |
|      | Router(config-sbc-sbe-cacpolicy-cactable-entry) # exit |         |
| 21   | **entry entry-id** | Enters the mode to create or modify an entry in an admission control table. |
|      | **Example:**     |         |
|      | Router(config-sbc-sbe-cacpolicy-cactable)# entry 2 |         |
| 22   | **match-value key** | Configures the match-value of an entry in a CAC Limit table.  
The `key` argument is a string used to match events. The format of the key is determined by the Limit table type (for example, Limit event-type tables or Limit call-priority tables). See the `match-value` command page for more details. |
|      | **Example:**     |         |
|      | Router(config-sbc-sbe-cacpolicy-cactable-entry)# match-value SIP-CUSTOMER-2 |         |
| 23   | **max-num-calls mnc** | Configures the maximum number of calls of an entry in an admission control table. |
|      | **Example:**     |         |
|      | Router(config-sbc-sbe-cacpolicy-cactable-entry)# max-num-calls 110 |         |
| 24   | **max-call-rate-per-scope limit [averaging-period period-num]** | Configures the maximum call rate for an entry in an admission control table.  
- `limit`—The limit for the number of new calls per minute. The value can range from 0 to 2147483647.  
- `averaging-period`—Specifies the averaging-period to use in the rate calculation. By default, 1 is selected.  
- `period-num`—Calculates rate-based on specified averaging period, ranging from 1 to 2. |
|      | **Example:**     |         |
|      | Router(config-sbc-sbe-cacpolicy-cactable-entry)# max-call-rate-per-scope 1000 averaging-period 2 |         |
### Command or Action | Purpose
---|---
**Step 25** max-bandwidth mbw bsize | Configures the maximum bidirectional bandwidth for an entry in an admission control table. For example, if a max-bandwidth value is configured, the SBC allows half of this value in each direction.

**Example:**
```
Router(config-sbc-sbe-cacpolicy-cactable-entry) # max-bandwidth 2000000 bps
```

The *mbw* argument is a positive integer specifying the total maximum rate at which call media should be admitted in both directions (in bytes per second).

The *bsize* argument specifies the transfer size to which *mbw* refers. Possible values are:
- bps
- Kbps
- Mbps
- Gbps

**Step 26** transcode-deny | Forbids transcoding for this entry in an admission control table.

**Example:**
```
Router(config-sbc-sbe-cacpolicy-cactable-entry) # transcode-deny
```

**Step 27** max-regs-rate-per-scope limit [averaging-period period-num] | Configures the maximum call number of subscriber registrations for an entry in an admission control table.

**Example:**
```
Router(config-sbc-sbe-cacpolicy-cactable-entry) # max-regs-rate-per-scope 300 averaging-period 2
```

- **limit**—The limit for the number of new calls per minute. The value can range from 0 to 2147483647.
- **averaging-period**—Specifies the averaging-period to use in the rate calculation. By default, 1 is selected.
- **period-num**—Calculates rate-based on specified averaging period, ranging from 1 to 2.

**Step 28** action [next-table goto-table-name | cac-complete] | Configures the action to perform after this entry in an admission control table. Possible actions are:

**Example:**
```
Router(config-sbc-sbe-cacpolicy-cactable-entry) # action cac-complete
```

- Identify the next CAC table to process using the *next-table* keyword and the *goto-table-name* argument.
- Stop processing for this scope using the *cac-complete* keyword.

**Step 29** exit | Exits from *entry* to *cactable* mode.

**Example:**
```
Router(config-sbc-sbe-cacpolicy-cactable-entry) # exit
```

**Step 30** exit | Exits from *cactable* to *cacpolicy* mode.

**Example:**
```
Router(config-sbc-sbe-cacpolicy-cactable) # exit
```
### How to Implement Policies

The following example shows the output of the **show sbc sbe cac-policy-set** command:

```console
Router# show sbc mySBC sbe cac-policy-set
SBC Service "mySBC"
CAC Averaging period 1: 100 sec
CAC Averaging period 2: 1500 sec

CAC Policy Set 2
Global policy set: Yes
First CAC table: 1
First CAC scope: src-adjacency

Table name: 1
Table type: limit adjacency
Total call setup failures (due to non-media limits): 0
Entry   Match value                         Action                  Failures
-----   -----------                         ------                  --------
1       SIPP1A                              Complete                       0
2       SIPP1B                              Complete                       0

CAC Policy Set 12
Global policy set: No
First CAC table: 1
First CAC scope: global

Table name: 1
Table type: limit adjacency
Total call setup failures (due to non-media limits): 0
```

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 31</strong></td>
<td>Completes the CAC policy set when you have committed the full set.</td>
</tr>
<tr>
<td>complete</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-cacpolicy)# complete</td>
<td></td>
</tr>
<tr>
<td><strong>Step 32</strong></td>
<td>Exits the SBE CAC policy mode.</td>
</tr>
<tr>
<td>exit</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-cacpolicy)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 33</strong></td>
<td>Activates the global CAC policy set. The CAC policy set must be in a complete state before it can be assigned as the default policy.</td>
</tr>
<tr>
<td>cac-policy-set global policy-num</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe)# cac-policy-set global 23</td>
<td></td>
</tr>
<tr>
<td><strong>Step 34</strong></td>
<td>Exits the SBE mode to Privileged EXEC mode.</td>
</tr>
<tr>
<td>end</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe)# end</td>
<td></td>
</tr>
<tr>
<td><strong>Step 35</strong></td>
<td>Displays details of the CAC policy sets configured on the SBC.</td>
</tr>
<tr>
<td>show sbc sbc-name sbe cac-policy-set [global]</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# show sbc mySBC sbe cac-policy-set</td>
<td></td>
</tr>
</tbody>
</table>

### Command or Action Purpose

- **complete**: Completes the CAC policy set when you have committed the full set.
- **exit**: Exits the SBE CAC policy mode.
- **cac-policy-set global policy-num**: Activates the global CAC policy set. The CAC policy set must be in a complete state before it can be assigned as the default policy.
  - **policy-num**: The call policy set number, ranging from 1 to 2147483647. The policy set must be in a complete state before it can be assigned as the default policy.
- **end**: Exits the SBE mode to Privileged EXEC mode.
- **show sbc sbc-name sbe cac-policy-set [global]**: Displays details of the CAC policy sets configured on the SBC.
  - **sbc-name**: Defines the name of the SBC service.
  - **global**: Lists the information pertaining to the global CAC policy set.
The following example shows the output of the `show sbc sbe cac-policy-set global` command:

```
Router# show sbc mySBC sbe cac-policy-set global
SBC Service "mySBC"
CAC Averaging period 1: 100 sec
CAC Averaging period 2: 1500 sec

CAC Policy Set 2
Global policy set: Yes
First CAC table: 1
First CAC scope: src-adjacency

Table name: 1
Table type: limit adjacency
Total call setup failures (due to non-media limits): 0
Entry   Match value                         Action                  Failures
-----   -----------                         ------                  --------
2       SIPP1B                              Complete                       0
1       SIPP1A                              Complete                       0
1       SIPP                                Complete                       0
```

The following example shows the output of the `show sbc sbe cac-policy-set global` command:

```
Router# show sbc mySBC sbe cac-policy-set global
SBC Service "mySBC"
CAC Averaging period 1: 100 sec
CAC Averaging period 2: 1500 sec

CAC Policy Set 2
Global policy set: Yes
First CAC table: 1
First CAC scope: src-adjacency

Table name: 1
Table type: limit adjacency
Total call setup failures (due to non-media limits): 0
Entry   Match value                         Action                  Failures
-----   -----------                         ------                  --------
2       SIPP1B                              Complete                       0
1       SIPP1A                              Complete                       0
1       SIPP                                Complete                       0
```

The following example shows the output of the `show sbc sbe cac-policy-set global` command:

```
Router# show sbc mySBC sbe cac-policy-set global
SBC Service "mySBC"
CAC Averaging period 1: 100 sec
CAC Averaging period 2: 1500 sec

CAC Policy Set 2
Global policy set: Yes
First CAC table: 1
First CAC scope: src-adjacency

Table name: 1
Table type: limit adjacency
Total call setup failures (due to non-media limits): 0
Entry   Match value                         Action                  Failures
-----   -----------                         ------                  --------
2       SIPP1B                              Complete                       0
1       SIPP1A                              Complete                       0
1       SIPP                                Complete                       0
```
Configuring Privacy Service

This section describes the tasks to configure the privacy service on a CAC policy set, adjacencies, and number analysis table:

- Configuring Privacy Service on a CAC Policy Set, page 7-126
- Configuring Privacy Service on Adjacencies, page 7-132
- Configuring a Number Analysis Table, page 7-134

Configuring Privacy Service on a CAC Policy Set

This task shows how to configure the privacy service on a CAC policy set.

**Note**
The `caller` and `callee` commands have been used in this procedure. In some scenarios, the `branch` command can be used as an alternative to the `caller` and `callee` command pair. The `branch` command has been introduced in Release 3.5.0. See the "Configuring Directed Nonlimiting CAC Policies? section on page 7-37 for information about this command.

**SUMMARY STEPS**

1. configure terminal
2. sbc `sbc-name`
3. sbe
4. cac-policy-set `policy-set-id`
5. cac-table `table-name`
6. table-type `{policy-set | limit {list of limit tables}}`
7. entry `entry-id`
8. caller-privacy edit-privacy-request `{pass | strip | insert | replace | sip {strip {all | critical | header | id | none | session | token word | user} | insert {critical | header | id | none | session | token word | user}}}
9. callee-privacy edit-privacy-request `{pass | strip | insert | replace | sip {strip {all | critical | header | id | none | session | token word | user} | insert {critical | header | id | none | session | token word | user}}}
10. caller-privacy privacy-service `{adj-trust-boundary | always | never}`
11. callee-privacy privacy-service `{adj-trust-boundary | always | never}`
12. end
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>configure terminal</td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>sbc sbc-name</td>
<td>Enters the SBC service mode. Use the <code>sbc-name</code> argument to define the name of the service.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config)# sbc mysbc</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>sbe</td>
<td>Enters the SBE entity mode within an SBC service.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>cac-policy-set policy-set-id</td>
<td>Enters the CAC policy set configuration mode within an SBE entity, creating a new policy set, if necessary.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe)# cac-policy-set 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>policy-set-id—The call policy set number that can range from 1 to 2147483647.</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>cac-table table-name</td>
<td>Enters the admission control table configuration mode (creating one, if necessary) within the context of an SBE policy set.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-cacpolicy)# cac-table StandardListByAccount</td>
<td></td>
</tr>
</tbody>
</table>
How to Implement Policies

Step 6

table-type {policy-set | limit (list of limit tables)}

Example:
Router(config-sbc-sbe-cacpolicy-cactable)#
table-type policy-set

Configures the table type of a CAC table within the context of an SBE policy set.
The list of limit tables argument controls the syntax of the match-value fields of the entries in the table. list of limit tables values are:

- `account` — Compares the name of the account.
- `adj-group` — Compares the name of the adjacency group.
- `adjacency` — Compares the name of the adjacency.
- `all` — No comparison type. All the events match this type.
- `call-priority` — Compares with call priority.
- `category` — Compares the number analysis-assigned category.
- `dst-account` — Compares the name of the destination account.
- `dst-adj-group` — Compares the name of the destination adjacency group.
- `dst-adjacency` — Compares the name of the destination adjacency.
- `dst-prefix` — Compares the beginning of the dialed digit string.
- `event-type` — Compares with CAC policy event types.
- `src-account` — Compares the name of the source account.
- `src-adj-group` — Compares the name of the source adjacency group.
- `src-adjacency` — Compares the name of the source adjacency.
- `src-prefix` — Compares the beginning of the calling number string.

Note For Limit tables, the event, message, or call matches only a single entry.

Features can be enabled or disabled per adjacency group through CAC configuration the same way this is done per individual adjacency. The adj-group table type matches on either the source adjacency group or the destination adjacency group.

After the policy-set keyword is specified, use the cac-scope command to configure the scope within each entry in which limits are applied in a CAC policy set table.

Note In Policy Set tables, the event, call, or message is applied to all the entries.
### Command or Action

#### Step 7

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>entry entry-id</strong></td>
<td>Enters the CAC table entry mode to create or modify an entry in an admission control table.</td>
</tr>
</tbody>
</table>

**Example:**

Router(config-sbc-sbe-cacpolicy-caetable)# entry 1
How to Implement Policies

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>caller-privacy edit-privacy-request (pass</td>
<td>Edits and updates the privacy indications provided by the user:</td>
</tr>
<tr>
<td>strip</td>
<td>insert</td>
</tr>
</tbody>
</table>

**Example:**

Router(config-sbc-sbe-cacpolicyc-cactable-entry) # caller-privacy edit-privacy-request strip

Step 8

To implement policies, you can use the following command structure:

- **pass** — Passes on the privacy header or the presentation indicators.
- **strip** — Removes all the privacy restrictions:
  - **SIP** — Removes the Privacy header.
  - **H323** — Sets the presentation indicator to allowed.
- **replace** — Replaces privacy restrictions:
  - **SIP** — Replaces Privacy:header;session;user;id;critical, except when none has been requested.
  - **H323** — Sets the presentation indicator to restricted.
- **insert** — Inserts privacy restrictions:
  - **SIP** — Inserts Privacy:header;session;user;id;critical if the header is not present already.
  - **H323** — Sets presentation indicator from allowed to restricted.
- **replace** — Inserts privacy tokens into the Privacy header.
- **strip** — Removes privacy tokens from the Privacy header.
- **critical** — Specifies the call to be discontinued if privacy cannot be achieved in the SIP Privacy header.
- **header** — Obscures all the header information, which is related to the user, from the SIP Privacy header.
- **id** — Removes ID headers from the SIP Privacy header.
- **none** — Privacy is not applied to the call.
- **session** — Specifies media privacy for the session in the SIP Privacy header. No media bypass is performed.
- **token** — Specifies the nonstandard user-defined privacy token in the SIP Privacy header.
- **word** — Specifies the user-defined privacy token.
- **user** — Removes all nonessential header information, which is related to the user, from the SIP Privacy header.

By default, the privacy setting value is set to **pass**.
### Command or Action

**Step 9**

```plaintext
callee-privacy edit-privacy-request {pass |
  strip | insert | replace | sip (strip |
  critical | header | id | none |
  session | token word |
  user} | insert (critical | header | id |
  none | session | token word |
  user)}
```

**Example:**

Router(config-sbc-sbe-cacpolicy-cactable-entry)# callee-privacy edit-privacy-request strip

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>pass</td>
<td>Edits and updates privacy indications provided by the user:</td>
</tr>
<tr>
<td>strip</td>
<td>- insert—Inserts privacy restrictions:</td>
</tr>
<tr>
<td></td>
<td>- SIP—Inserts Privacy:header;session;user;id;critical if the header is not present already.</td>
</tr>
<tr>
<td></td>
<td>- H323—Sets presentation indicator from allowed to restricted.</td>
</tr>
<tr>
<td>replace</td>
<td>- sip—Specifies the following SIP settings. This allows greater control and overrides all generic actions:</td>
</tr>
<tr>
<td></td>
<td>- insert—Inserts privacy tokens into the Privacy header.</td>
</tr>
<tr>
<td></td>
<td>- strip—Removes privacy tokens from the Privacy header.</td>
</tr>
<tr>
<td></td>
<td>- critical—Specifies the call to be discontinued if privacy cannot be achieved in the SIP Privacy header.</td>
</tr>
<tr>
<td></td>
<td>- header—Obscures all the header information, which is related to the user, from the SIP Privacy header.</td>
</tr>
<tr>
<td></td>
<td>- id—Removes ID headers from the SIP Privacy header.</td>
</tr>
<tr>
<td></td>
<td>- none—Privacy is not applied to the call.</td>
</tr>
<tr>
<td></td>
<td>- session—Specifies media privacy for the session in the SIP Privacy header. No media bypass is performed.</td>
</tr>
<tr>
<td></td>
<td>- token—Specifies the nonstandard user-defined privacy token in the SIP Privacy header.</td>
</tr>
<tr>
<td></td>
<td>- word—Specifies the user-defined privacy token.</td>
</tr>
<tr>
<td></td>
<td>- user—Removes all nonessential header information, which is related to the user, from the SIP Privacy header.</td>
</tr>
</tbody>
</table>

By default, the privacy setting value is set to **pass**.
### How to Implement Policies

#### Configuring Privacy Service on Adjacencies

This task shows how to configure the privacy service on the SIP and H323 adjacencies.

### SUMMARY STEPS

1. `configure terminal`
2. `sbc sbc-name`
3. `sbe`
4. `adjacency sip adjacency-name`
5. `privacy [inherit-profile | trusted | untrusted]`
6. `exit`
7. `adjacency h323 adjacency-name`

---

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 10</strong></td>
<td>Configures privacy settings according to RFC3323, RFC3325, and/or setting of the H.323 presentation restriction settings in a given entry in the admission control table:</td>
</tr>
<tr>
<td>`caller-privacy privacy-service (adj-trust-boundary</td>
<td>always</td>
</tr>
</tbody>
</table>
| **Example:**      | Router(config-sbc-sbe-cacpolicy-cactable-entry)  
# caller-privacy privacy-service always |

By default, the privacy setting value is set to `adj-trust-boundary`.

<table>
<thead>
<tr>
<th>Step 11</th>
<th>Configures privacy settings according to RFC3323, RFC3325, and/or setting of H.323 presentation restriction settings in a given entry in the admission control table:</th>
</tr>
</thead>
<tbody>
<tr>
<td>`callee-privacy privacy-service (adj-trust-boundary</td>
<td>always</td>
</tr>
</tbody>
</table>
| **Example:** | Router(config-sbc-sbe-cacpolicy-cactable-entry)  
# callee-privacy privacy-service adj-trust-boundary |

By default, the privacy setting value is set to `adj-trust-boundary`.

<table>
<thead>
<tr>
<th>Step 12</th>
<th>Exits from the CAC table entry configuration mode and enters the Privileged EXEC mode.</th>
</tr>
</thead>
</table>
| `end`   | Example: | Router(config-sbc-sbe-cacpolicy-cactable-entry)  
# end |

---

**Step 10**

**SUMMARY STEPS**

1. `configure terminal`
2. `sbc sbc-name`
3. `sbe`
4. `adjacency sip adjacency-name`
5. `privacy [inherit-profile | trusted | untrusted]`
6. `exit`
7. `adjacency h323 adjacency-name`
8. allow private info
9. end

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc <strong>sbc-name</strong></td>
<td>Enters the SBC service mode. Use the <em>sbc-name</em> argument to define the name of the service.</td>
</tr>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Enters the SBE entity mode within an SBC service.</td>
</tr>
<tr>
<td><strong>Step 4</strong> adjacency sip <strong>adjacency-name</strong></td>
<td>Enters the SBE SIP adjacency mode. Use the <em>adjacency-name</em> argument to define the name of the service.</td>
</tr>
<tr>
<td><strong>Step 5</strong> privacy **[inherit-profile</td>
<td>trusted</td>
</tr>
<tr>
<td>Example: Router(config-sbe-adj-sip)# privacy trusted</td>
<td>• <em>inherit-profile</em>—Specifies that the trust level for determining whether privacy services are required is derived from the adjacencies inherit-profile.</td>
</tr>
<tr>
<td>Example: Router(config-sbe-adj-sip)# exit</td>
<td>• <em>trusted</em>—Specifies that the adjacency is trusted and privacy services do not have to be applied.</td>
</tr>
<tr>
<td>Example: Router(config-sbe-adj-sip)# adjacency h323 test</td>
<td>• <em>untrusted</em>—Specifies that the adjacency is not trusted and requires privacy services to be applied.</td>
</tr>
<tr>
<td><strong>Step 6</strong> exit</td>
<td>Exits the SIP adjacency mode and enters the SBE mode.</td>
</tr>
<tr>
<td>Example: Router(config-sbe-adj-sip)# exit</td>
<td>By default, the trust level is set to <em>inherit-profile</em>.</td>
</tr>
<tr>
<td><strong>Step 7</strong> adjacency h323 <strong>adjacency-name</strong></td>
<td>Configures a destination H.323 adjacency for the SBC service, and enters into H. 323 adjacency configuration mode.</td>
</tr>
<tr>
<td>Example: Router(config-sbc-sbe)# adjacency h323 test</td>
<td>A destination H.323 adjacency is where the configured failure return codes cause hunting to occur. This command overrides any globally configured retry error codes.</td>
</tr>
</tbody>
</table>
How to Implement Policies

Configuring a Number Analysis Table

This task shows how to configure a number analysis table to detect anonymity.

SUMMARY STEPS

1. configure terminal
2. sbc sbc-name
3. sbe
4. call-policy-set policy-set-id
5. na-src-name-anonymous-table table-name
6. entry entry-id
7. match-anonymous [false | true]
8. end

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 8 allow private info</td>
<td>Configures the H.323 adjacency to allow private information to be sent.</td>
</tr>
<tr>
<td>Example: Router(config-sbe-adj-h323)# allow private info</td>
<td>By default, the H.323 adjacency does not send the private information of a user.</td>
</tr>
<tr>
<td>Step 9 end</td>
<td>Exits from a H.323 adjacency configuration mode and enter the Privileged EXEC mode.</td>
</tr>
<tr>
<td>Example: Router(config-sbe-adj-h323)# end</td>
<td></td>
</tr>
</tbody>
</table>
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc sbc-name</td>
<td>Enters the SBC service mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Use the <code>sbc-name</code> argument to define the name of the service.</td>
</tr>
<tr>
<td>Router(config)# sbc mySbc</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Enters the SBE entity mode within an SBC service.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> call-policy-set policy-set-id</td>
<td>Enters the routing policy set configuration mode within an SBE entity.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe)# call-policy-set 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> na-src-name-anonymous-table table-name</td>
<td>Enters the configuration mode of a number analysis table to</td>
</tr>
<tr>
<td>Example:</td>
<td>determine whether the display name or presentation number is</td>
</tr>
<tr>
<td>Router(config-sbc-sbe-rtgpolicy)# na-src-name-anonymous-table NameTable</td>
<td>anonymous.</td>
</tr>
<tr>
<td><strong>Step 6</strong> entry entry-id</td>
<td>Enters the number analysis table entry mode for configuring an entry</td>
</tr>
<tr>
<td>Example:</td>
<td>in a number analysis table, creating the entry, if necessary.</td>
</tr>
<tr>
<td>Router(config-sbc-sbe-rtgpolicy-natable)# entry</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> match-anonymous [false</td>
<td>true]</td>
</tr>
<tr>
<td>Example:</td>
<td>na-src-name-anonymous-table number analysis table.</td>
</tr>
<tr>
<td>Router(config-sbc-sbe-rtgpolicy-natable-entry)# match-anonymous false</td>
<td>• false—Specifies the display name or presentation number as not</td>
</tr>
<tr>
<td></td>
<td>• true—Specifies the display name or presentation number as anonymous.</td>
</tr>
<tr>
<td><strong>Step 8</strong> end</td>
<td>Exits the number analysis table entry mode and enters the Privileged</td>
</tr>
<tr>
<td>Example:</td>
<td>EXEC mode.</td>
</tr>
<tr>
<td>Router(config-sbc-sbe-rtgpolicy-natable-entry)# end</td>
<td></td>
</tr>
</tbody>
</table>
Configuring Multiple SBC Media Bypass

This task shows how to configure the Multiple SBC Media Bypass feature. The steps to configure the renegotiation of media bypass after a session refreshes are also included in this task.

Note

The caller and callee commands have been used in this procedure. In some scenarios, the branch command can be used as an alternative to the caller and callee command pair. The branch command has been introduced in Release 3.5.0. See the Configuring Directed Nonlimiting CAC Policies? section on page 7-37 for information about this command.

SUMMARY STEPS

1. configure terminal
2. sbc sbc-name
3. sbe
4. adjacency sip adjacency-name
5. media bypass {max-data-len data-length | tag sequence-number tag-name}
6. exit
7. cac-policy-set policy-set-id
8. cac-table table-name
9. table-type {policy-set | limit {list of limit tables}}
10. entry entry-id
11. match-value key
12. media bypass type [all | none | full | hairpin partial] | hairpin [full partial] | partial [full hairpin]
13. caller media bypass {enable | disable}
14. callee media bypass {enable | disable}
15. action [next-table goto-table-name | cac-complete]
16. exit
17. entry entry-id
18. session-refresh renegotiation {allow | suppress}
19. end
20. show sbc sbc-name sbe cac-policy-set id table name entry entry
21. show sbc sbc-name sbe adjacencies adjacency-name detail
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>configure terminal</code></td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>sbc</code> <code>sbc-name</code></td>
<td>Enters the SBC service mode. Use the <code>sbc-name</code> argument to define the name of the service.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# sbc mySBC</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>sbe</code></td>
<td>Enters the SBE entity mode within an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> <code>adjacency sip</code> <code>adjacency-name</code></td>
<td>Enters the SBE SIP adjacency mode. Use the <code>adjacency-name</code> argument to define the name of the service.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe)# adjacency sip access</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> <code>media bypass</code> (<code>max-data-len</code> <code>data-length</code></td>
<td>Configures the multiple SBC media bypass feature on a SIP adjacency:</td>
</tr>
<tr>
<td></td>
<td><code>tag</code> <code>sequence-number</code> <code>tag-name</code>)</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-adj-sip)# media bypass tag 1 TAG1</td>
<td></td>
</tr>
</tbody>
</table>

**Note** Media bypass is not supported for H.323 calls.
### How to Implement Policies

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td><code>exit</code></td>
<td>Exits the adjacency SIP mode and enters the SBE entity mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Router(config-sbc-sbe-adj-sip)# exit</code></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td><code>cac-policy-set policy-set-id</code></td>
<td>Enters the CAC policy set configuration mode within an SBE entity, creating a new policy set if necessary.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Router(config-sbc-sbe)# cac-policy-set 1</code></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td><code>cac-table table-name</code></td>
<td>Enters the admission control table configuration mode (creating one if necessary) within the context of an SBE policy set.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Router(config-sbc-sbe-cacpolicy)# cac-table MyTable</code></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>`table-type {policy-set</td>
<td>limit (list of limit tables)}`</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Router(config-sbc-sbe-cacpolicy-cactable)# table-type src-adjacency</code></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td><code>entry entry-id</code></td>
<td>Enters the mode to create or modify an entry in an admission control table.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Router(config-sbc-sbe-cacpolicy-cactable)# entry 1</code></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td><code>match-value key</code></td>
<td>Configures the match-value of an entry in a CAC Limit Table.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Router(config-sbc-sbe-cacpolicy-cactable-entry)# match-value access</code></td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

**Step 12**

```plaintext
media bypass type [all | none | full [hairpin partial] | hairpin [full partial] | partial [full hairpin]]
```

**Example:**
```
Router(config-sbc-sbe-cacpolicy-cactable-entry)
# media bypass type full hairpin
```

**Purpose:**
Configures the multiple SBC media bypass feature for CAC policy set.
- **all**—Enables all, such as partial, hairpin, and full types of media bypass for the CAC table entry.
- **none**—Disables all types of media bypass for the CAC table entry.
- **full**—Enables media bypass on the SBC if adjacent and non-adjacent downstream and upstream hops have direct media connectivity, common tags in bypass tag list or with same VPN.
- **hairpin**—Enables media bypass for the hairpin calls.
- **partial**—Enables media bypass if the SBC is a member of a group of SBCs that share the same IP realm and if even one SBC within that group is on the media path.

**Note:** If the media bypass type is explicitly configured to be partial, only IP realm and VPN configuration on the adjacency can be used to determine whether media bypass is possible. Because media bypass tags are not used, the VPN names must be globally unique across all the SBCs for partial media bypass to work.

**Step 13**

```plaintext
caller media bypass {enable | disable}
```

**Example:**
```
Router(config-sbc-sbe-cacpolicy-cactable-entry)
# caller media bypass enable
```

**Purpose:**
Enables or disables the multiple SBC media bypass feature on the caller side.
- **enable**—Enables the multiple SBC media bypass feature on the caller side.
- **disable**—Disables the multiple SBC media bypass feature on the caller side.

**Step 14**

```plaintext
callee media bypass {enable | disable}
```

**Example:**
```
Router(config-sbc-sbe-cacpolicy-cactable-entry)
# callee media bypass enable
```

**Purpose:**
Enables or disables the multiple SBC media bypass feature on the callee side.
- **enable**—Enables the multiple SBC media bypass feature on the callee side.
- **disable**—Disables the multiple SBC media bypass feature on the callee side.

**Step 15**

```plaintext
action [next-table goto-table-name | cac-complete]
```

**Example:**
```
Router(config-sbc-sbe-cacpolicy-cactable-entry)
# action cac-complete
```

**Purpose:**
Configures the action to be performed after this entry, in an admission control table. Possible actions are:
- Identify the CAC table to be processed next using the `next-table` keyword and the `goto-table-name` argument.
- Stop the processing action for this scope using the `cac-complete` keyword.
### How to Implement Policies

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| 16   | `exit`            | Configures the action to be performed after this entry, in an admission control table. Possible actions are:  
- Identify the CAC table to be processed next using the `next-table` keyword and the `goto-table-name` argument.  
- Stop the processing action for this scope using the `cac-complete` keyword. |
| 17   | `entry entry-id` | Enters the mode to create or modify an entry in an admission control table. |
| 18   | `session-refresh renegotiation {allow | suppress}` | Depending on the option that you select, one of the following actions is configured:  
- **allow**—Specifies that an offer that contains duplicate SDP must be processed using the normal offer-answer rules. Media reservations can change, and interworking functions can be renegotiated.  
- **suppress**—Specifies that an offer that contains duplicate SDP must be processed using the session refresh variant of the offer-answer rules. Media reservations are not changed, and interworking functions are not renegotiated. The SBC forwards the last sent offer or answer regardless of the offer or answer that was received.  
The default is that the session refresh strategy for the call is not affected by this CAC policy entry. |
| 19   | `end`             | Exits from the CAC table entry configuration mode and enters the Privileged EXEC mode. |
| 20   | `show sbc sbc-name sbe cac-policy-set id table name entry entry` | Displays detailed information about a specific entry in a CAC policy table. |
| 21   | `show sbc sbc-name sbe adjacencies adjacency-name detail` | Displays all the detailed field outputs for the specified SIP adjacency. |
Configuring Common IP Address Media Bypass

This procedure shows how to configure the Common IP Address Media Bypass feature.

### SUMMARY STEPS

1. configure terminal
2. sbc sbc-name
3. sbe
4. adjacency sip adjacency-name
5. media bypass auto-nat-tag-gen
6. end
7. show sbc sbc-name sbe adjacencies adjacency-name detail

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc sbc-name</td>
<td>Enters the mode of an SBC service. Use the <em>sbc-name</em> argument to define the name of the service.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# sbc mysbc</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Enters the mode of an SBE entity within an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> adjacency sip adjacency-name</td>
<td>Enters the mode of an SBE SIP adjacency.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe)# adjacency sip access-side-1</td>
<td>• <em>adjacency-name</em>—Name of the adjacency.</td>
</tr>
<tr>
<td><strong>Step 5</strong> media bypass auto-nat-tag-gen</td>
<td>Configures the Common IP Address Media Bypass feature to generate a media bypass tag for the registered endpoints that are behind a NAT device associated with this adjacency.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-adj-sip)# media bypass auto-nat-tag-gen</td>
<td></td>
</tr>
</tbody>
</table>
How to Implement Policies

Activating a CAC Policy Set

This task activates a global CAC policy set.

SUMMARY STEPS

1.  configure terminal
2.  sbc sbc-name
3.  sbe
4.  cac-policy-set global policy-set-id

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 6  configure terminal</td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router# configure terminal</td>
</tr>
<tr>
<td>Step 2  sbc sbc-name</td>
<td>Enters the mode of an SBC service. Use the sbc-name argument to define</td>
</tr>
<tr>
<td>Example:</td>
<td>the name of the service.</td>
</tr>
<tr>
<td>Step 3  sbe</td>
<td>Enters the mode of an SBE entity within an SBC service.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config)# sbe mysbc</td>
</tr>
<tr>
<td>Step 4  cac-policy-set global policy-set-id</td>
<td>Activates the global CAC policy set within an SBE entity.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe)# cac-policy-set global 1</td>
</tr>
</tbody>
</table>

    • policy-set-id—The call policy set number that can range from 1 to 2147483647.
## Configuring Asymmetric Payload Types

This task configures SBC to allow asymmetric payload types.

### SUMMARY STEPS

1. `configure terminal`
2. `sbc sbc-name`
3. `sbe`
4. `cac-policy-set policy-set-id`
5. `first-cac-table table-name`
6. `cac-table table-name`
7. `table-type policy-set`
8. `entry entry-id`
9. `action cac-complete`
10. `payload-type asymmetric allowed`
11. `complete`
12. `cac-policy-set global policy-set-id`
13. `end`
14. `show sbc sbc-name sbe cac-policy-set`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td></td>
</tr>
<tr>
<td><code>configure</code></td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router# configure</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td></td>
</tr>
<tr>
<td><code>sbc sbc-name</code></td>
<td>Enables entry into the mode of an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config)# sbc mySbc</code></td>
<td>Use the <code>sbc-name</code> argument to define the name of an SBC.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td></td>
</tr>
<tr>
<td><code>sbe</code></td>
<td>Enables entry into the mode of an SBE entity within an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-sbc)# sbe</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td></td>
</tr>
<tr>
<td><code>cac-policy-set policy-set-id</code></td>
<td>Enables entry into the mode of the CAC policy.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-sbc-sbe)# cac-policy-set 1</code></td>
<td></td>
</tr>
</tbody>
</table>
### Configuring Asymmetric Payload Types

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 5</strong> first-cac-table table-name</td>
<td>Configures the name of the first policy table to be processed when performing the admission control stage of the CAC policy.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-cacpolicy)# first-cac-table first_policy_table</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> cac-table table-name</td>
<td>Enables entry into the mode for configuring an admission control table (or creating one, if necessary) within the context of an SBE policy set.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-cacpolicy)# cac-table first_policy_table</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> table-type {policy-set</td>
<td>limit {list of limit tables}}</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-cacpolicy-cactable)# table-type policy-set</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> entry entry-id</td>
<td>Enables entry into the mode to create or modify an entry in an admission control table.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-cacpolicy-cactable)# entry 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong> action [next-table goto-table-name</td>
<td>cac-complete]</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-cacpolicy-cactable-entry)# action cac-complete</td>
<td></td>
</tr>
<tr>
<td><strong>Step 10</strong> payload-type asymmetric allowed</td>
<td>Configures SBC to allow asymmetric payload types.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-cacpolicy-cactable-entry)# payload-type asymmetric allowed</td>
<td></td>
</tr>
<tr>
<td><strong>Step 11</strong> complete</td>
<td>Completes the CAC policy.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-cacpolicy)# complete</td>
<td></td>
</tr>
<tr>
<td><strong>Step 12</strong> cac-policy-set global policy-set-id</td>
<td>Activates the global CAC policy set within an SBE entity.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe)# cac-policy-set global 1</td>
<td></td>
</tr>
</tbody>
</table>
New router features, such as transcoding, transrating, and inband DTMF interworking, have been introduced in earlier releases. If no limits are set on the number of calls that use the resources provided by these features, overload conditions may occur and the router may stop responding. You can configure limits on resource usage to prevent the occurrence of overload conditions. This is one of the areas in which Cisco Unified Border Element (SP Edition) policies can be applied.

The Limiting Resource Usage feature has been introduced in Release 3.4S.

You can configure media policies to specify maximum levels of usage for the following:

- Number of audio streams using transcoding
- Number of audio streams using transrating
- Number of video streams using transcoding
- Number of audio streams using inband DTMF interworking
- Number of streams using SRTP encryption and decryption
- Number of registered subscribers using IPSec encryption and decryption on the signaling link to the SBC
- Number of calls made by subscribers who are using IPSec-protected signaling
- Total number of video and audio streams using transcoding, transrating, inband DTMF interworking, and SRTP encryption and decryption—weighted by the costs assigned to each of these resources.

Table 7-10 lists the default resource costs. You can modify these default resource costs.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Default Resource Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audio transcoding</td>
<td>10</td>
</tr>
<tr>
<td>Audio transrating</td>
<td>6</td>
</tr>
<tr>
<td>Video transcoding</td>
<td>50</td>
</tr>
</tbody>
</table>

**Note**

The Limiting Resource Usage feature has been introduced in Release 3.4S.

**Step 13**

```
end
```

Example:

```
Router(config-sbc-sbe-cacpolicy)# end
```

**Purpose**

Enables exit from the CAC policy set configuration mode and entry into the Privileged EXEC mode.

**Step 14**

```
show sbc sbc-name sbe cac-policy-set id table name entry entry
```

Example:

```
Router# show sbc mysbc sbe cac-policy-set 1 table standard_policy_list entry 1
```

**Purpose**

Displays detailed information for a specific entry in a CAC policy table, including any restricted codecs.

**Table 7-10**  **Default Resource Costs**

<table>
<thead>
<tr>
<th>Resource</th>
<th>Default Resource Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audio transcoding</td>
<td>10</td>
</tr>
<tr>
<td>Audio transrating</td>
<td>6</td>
</tr>
<tr>
<td>Video transcoding</td>
<td>50</td>
</tr>
</tbody>
</table>
Limiting Resource Usage

Chapter 7  Implementing Cisco Unified Border Element (SP Edition) Policies

Chapter 7      Implementing Cisco Unified Border Element (SP Edition) Policies

Limiting Resource Usage

At run time, the total resource usage value is calculated for each incoming call using the resource costs configured for the resources requested by the call. This calculated value is then compared with the maximum total resource usage value that you have configured. If the calculated value is more than the configured value, the media policy rejects the call. This means that the call either fails or is directed to a different message gateway or signaling route.

After you define a media policy, you can apply it in one of the following ways:

- As a CAC policy
  For example, call-scoped policies restrict resource usage for a particular call. In contrast, adjacency-scoped policies restrict resource usage at the adjacency level.

- As a media gateway policy
  Media policies applied at the media gateway level restrict resource usage for the media gateway.

After you apply a media policy, you can view the resource usage of each resource for which you have specified a limit in the media policy. For example, you can view the number of media streams that are being video-transcoded by the message gateway on which you have applied the media policy.

The following sections describe the procedures for limiting resource usage:

- Configuring Resource Costs for Transcoding, Transrating, Inband DTMF Interworking, and SRTP Encryption and Decryption, page 7-146
- Configuring Usage Limits for Transcoding, Transrating, Inband DTMF Interworking, and SRTP Encryption and Decryption, page 7-149
- Configuring Usage Limits for IPSec Encryption and Decryption and IPSec-Protected Signaling, page 7-153
- Example: Limiting Resource Usage, page 7-167

Configuring Resource Costs for Transcoding, Transrating, Inband DTMF Interworking, and SRTP Encryption and Decryption

The resource costs that you have configured are used to calculate and compare the total weighted resource usage against the maximum total usage that you have configured. Table 7-10 shows the default resource costs. You can modify these resource costs to suit the requirements of your operating environment.

This task explains how to configure resource costs for transcoding, transrating, inband DTMF interworking, and SRTP encryption and decryption.

**SUMMARY STEPS**

1. `configure terminal`
2. `sbc sbc-name`
3. `sbe`

<table>
<thead>
<tr>
<th>Table 7-10 Default Resource Costs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource</td>
</tr>
<tr>
<td>---------------------------------</td>
</tr>
<tr>
<td>Inband DTMF interworking</td>
</tr>
<tr>
<td>SRTP encryption and decryption</td>
</tr>
</tbody>
</table>

At run time, the total resource usage value is calculated for each incoming call using the resource costs configured for the resources requested by the call. This calculated value is then compared with the maximum total resource usage value that you have configured. If the calculated value is more than the configured value, the media policy rejects the call. This means that the call either fails or is directed to a different message gateway or signaling route.

After you define a media policy, you can apply it in one of the following ways:

- As a CAC policy
  For example, call-scoped policies restrict resource usage for a particular call. In contrast, adjacency-scoped policies restrict resource usage at the adjacency level.

- As a media gateway policy
  Media policies applied at the media gateway level restrict resource usage for the media gateway.

After you apply a media policy, you can view the resource usage of each resource for which you have specified a limit in the media policy. For example, you can view the number of media streams that are being video-transcoded by the message gateway on which you have applied the media policy.

The following sections describe the procedures for limiting resource usage:

- Configuring Resource Costs for Transcoding, Transrating, Inband DTMF Interworking, and SRTP Encryption and Decryption, page 7-146
- Configuring Usage Limits for Transcoding, Transrating, Inband DTMF Interworking, and SRTP Encryption and Decryption, page 7-149
- Configuring Usage Limits for IPSec Encryption and Decryption and IPSec-Protected Signaling, page 7-153
- Example: Limiting Resource Usage, page 7-167

Configuring Resource Costs for Transcoding, Transrating, Inband DTMF Interworking, and SRTP Encryption and Decryption

The resource costs that you have configured are used to calculate and compare the total weighted resource usage against the maximum total usage that you have configured. Table 7-10 shows the default resource costs. You can modify these resource costs to suit the requirements of your operating environment.

This task explains how to configure resource costs for transcoding, transrating, inband DTMF interworking, and SRTP encryption and decryption.

**SUMMARY STEPS**

1. `configure terminal`
2. `sbc sbc-name`
3. `sbe`
4. media-gateway policy type {default | local | remote {ipv4 | ipv6} ip-address [port port-number]]}
5. transcode audio cost number
6. transcode video cost number
7. transrate audio cost number
8. interwork inband-dtmf cost number
9. interwork srtp cost number
10. end
11. show sbc sbc-name sbe media-gateway-policy
12. show sbc sbc-name sbe media-gateway-policy [stats | type {default | local | remote {ipv4 | ipv6} ip-address [port port-number]]]

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1**
  configure terminal | Enables the global configuration mode. |
| **Example:**
  Router# configure terminal | |
| **Step 2**
  sbc sbc-name | Enters the SBC service mode. |
| **Example:**
  Router(config)# sbc mySbc | • sbc-name—Name of the SBC. |
| **Step 3**
  sbe | Enters the SBE configuration mode. |
| **Example:**
  Router(config-sbc)# sbe | |
Limiting Resource Usage

Chapter 7  Implementing Cisco Unified Border Element (SP Edition) Policies

Command or Action | Purpose
--- | ---
media-gateway policy type {default | local | (remote (ipv4 | ipv6) ip-address [port port-number])} | Configures a media gateway policy.
  - **default**—Specifies that the media gateway policy must be applied to all media gateways configured on the SBC. A default media gateway policy is applied on a media gateway (local or remote) when no other media policy is applied on the media gateway.
  - **local**—Specifies that the media gateway policy must be applied to the media gateway that is locally configured on the SBC.
  - **remote**—Specifies that the media gateway policy must be applied to a remote media gateway.
  - **ipv4**—Specifies that the remote media gateway has an IPv4 IP address.
  - **ipv6**—Specifies that the remote media gateway has an IPv6 IP address.
  - **ip-address**—IP address of the remote media gateway.
  - **port**—Specifies the port number of the remote media gateway.
  - **port-number**—Port number of the remote media gateway.

Example:
Router(config-sbc-sbe)# media-gateway policy type remote ipv4 192.0.2.26 6886

Step 5
transcode audio cost number | Specifies the resource cost for transcoding an audio stream.
  - **number**—Resource cost. The range is from 1 to 4294967295. As mentioned in Table 7-10, the default cost is 10.

Example:
Router(config-sbc-sbe-media-pol)# transcode audio cost 10

Step 6
transcode video cost number | Specifies the resource cost for transcoding a video stream.
  - **number**—Resource cost. The range is from 1 to 4294967295. As mentioned in Table 7-10, the default cost is 50.

Example:
Router(config-sbc-sbe-media-pol)# transcode video cost 55

Step 7
transrate audio cost number | Specifies the resource cost for transrating an audio stream.
  - **number**—Resource cost. The range is from 1 to 4294967295. As mentioned in Table 7-10, the default cost is 6.

Example:
Router(config-sbc-sbe-media-pol)# transrate audio cost 10

Step 8
interwork inband-dtmf cost number | Specifies the resource cost for an audio stream using inband DTMF interworking.
  - **number**—Resource cost. The range is from 1 to 4294967295. As mentioned in Table 7-10, the default cost is 4.

Example:
Router(config-sbc-sbe-media-pol)# interwork inband-dtmf cost 6
Limiting Resource Usage

The following example shows the output of the `show sbc sbe media-gateway-policy type` command for a specified media gateway policy:

```
Router# show sbc mySbc sbe media-gateway-policy type remote ipv4 192.0.2.26 port 6886
```

Gateway Policy Type = REMOTE

Remote vpn = 0
Remote address type = IPV4
Remote address = 192.0.2.26
Remote Port = 6886
Media Limit Table =
Transcode Audio Cost = 10
Transrate Audio Cost = 6

Configuring Usage Limits for Transcoding, Transrating, Inband DTMF Interworking, and SRTP Encryption and Decryption

This task describes how to configure usage limits for transcoding, transrating, inband DTMF interworking, and SRTP encryption and decryption.
Chapter 7  Implementing Cisco Unified Border Element (SP Edition) Policies

Limiting Resource Usage

SUMMARY STEPS

1. configure terminal
2. sbc sbc-name
3. sbe
4. media-policy policy-name
5. type {cac-policy | gateway}
6. transcode audio maximum number
7. transcode video maximum number
8. transrate audio maximum number
9. interwork inband-dtmf maximum number
10. interwork srtp maximum number
11. total resource maximum number
12. exit
13. media-gateway policy type {default | local | [remote {ipv4 | ipv6} ip-address [port port-number]]}
14. media limits policy-name
15. end
16. show sbc sbc-name sbe media-policy
17. show sbc sbc-name sbe media-policy policy-name

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router# configure terminal</td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc sbc-name</td>
<td>Enters the SBC service mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config)# sbc mySbc</td>
</tr>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Enters the SBE configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc)# sbe</td>
</tr>
<tr>
<td><strong>Step 4</strong> media-policy policy-name</td>
<td>Specifies the media policy to be created.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc)# media-policy media_policy2</td>
</tr>
</tbody>
</table>
### Command or Action

| Step 5 | type {cac-policy | gateway} | Specifies the type of media policy table to be configured. You can specify one of the following media policy types: |
|--------|-----------------------------|-------------------------------------------------|
| Example: | Router(config-sbc-sbe-media-pol)# type gateway | • cac-policy—Specifies that a media policy table must be configured for a CAC-policy type policy.  
• gateway—Specifies that a media policy table must be configured for a gateway type policy. |

<table>
<thead>
<tr>
<th>Step 6</th>
<th>transcode audio maximum number</th>
<th>Specifies the maximum number of media streams that can be audio transcoded at any point of time.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-media-pol)# transcode audio maximum 20000</td>
<td>• number—Number of media streams. The range is from 1 to 4294967295. The default is 4294967295.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 7</th>
<th>transcode video maximum number</th>
<th>Specifies the maximum number of media streams that can be video transcoded at any point of time.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-media-pol)# transcode video maximum 20000</td>
<td>• number—Number of media streams. The range is from 1 to 4294967295. The default is 4294967295.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 8</th>
<th>transrate audio maximum number</th>
<th>Specifies the maximum number of media streams that can be audio transrated at any point of time.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-media-pol)# transrate audio maximum 6000</td>
<td>• number—Number of media streams. The range is from 1 to 4294967295. The default is 4294967295.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 9</th>
<th>interwork inband-dtmf maximum number</th>
<th>Specifies the maximum number of media streams that can use the inband DTMF interworking resource at any point of time.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-media-pol)# interwork inband-dtmf maximum 2000</td>
<td>• number—Number of media streams. The range is from 1 to 4294967295. The default is 4294967295.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 10</th>
<th>interwork srtp maximum number</th>
<th>Specifies the maximum number of media streams that can use the SRTP interworking resource at any point of time.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-media-pol)# interwork srtp maximum 500</td>
<td>• number—Number of media streams. The range is from 1 to 4294967295. The default is 4294967295.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 11</th>
<th>total resource maximum number</th>
<th>Specifies the total number of video and audio streams that can use transcoding, transrating, inband DTMF interworking, and SRTP encryption and decryption—weighted by the costs assigned to each of these resources.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-media-pol)# total resource maximum 35000</td>
<td>• number—Number of media streams. The range is from 1 to 4294967295. The default is 4294967295.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 12</th>
<th>exit</th>
<th>Exits the SBE media policy configuration mode, and enters the SBE configuration mode.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-media-pol)# exit</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 13** media-gateway policy type {default | local | (remote {ipv4 | ipv6} ip-address [port port-number])} | Configures a media gateway policy.  
- **default**—Specifies that the media gateway policy must be applied to all media gateways configured on the SBC. A default media gateway policy is applied on a media gateway (local or remote) when no other media policy is applied on the media gateway.  
- **local**—Specifies that the media gateway policy must be applied to the media gateway that is locally configured on the SBC.  
- **remote**—Specifies that the media gateway policy must be applied to a remote media gateway.  
- **ipv4**—Specifies that the remote media gateway has an IPv4 IP address.  
- **ipv6**—Specifies that the remote media gateway has an IPv6 IP address.  
- **ip-address**—IP address of the remote media gateway. The IP address can be in the IPv4 format or IPv6 format.  
- **port**—Specifies the port number of the remote media gateway.  
- **port-number**—Port number of the remote media gateway. |
| **Example:** Router(config-sbc-sbe)# media-gateway policy type default | |

### Command or Action

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 14** media limits policy-name | Specifies the media policy to be associated with the CAC policy table entry or applied on the media gateway.  
- **policy-name**—Name of the media policy. |
| **Example:** Router(config-sbc-sbe-mg-pol)# media limits media_policy2 | |

### Command or Action

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 15</strong> end</td>
<td>Exits the media policy configuration mode, and enters the privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-media-pol)# end</td>
<td></td>
</tr>
</tbody>
</table>

### Command or Action

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 16** show sbc sbc-name sbe media-policy | Displays details of all media policies. These details include the resource usage limits that you have configured.  
- **sbc-name**—Name of the SBC service. |
| **Example:** Router# show sbc mySbc sbe media-policy | |

### Command or Action

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 17** show sbc sbc-name sbe media-policy policy-name | Displays details of all media policies. These details include the resource usage limits that you have configured.  
- **sbc-name**—Name of the SBC service.  
- **policy-name**—Name of the media policy. |
| **Example:** Router# show sbc mySbc sbe media-policy | |
The following example shows the output of the `show sbc sbe media-policy` command for a specified media policy:

```
Router# show sbc mySbc sbe media-policy my_media_policy
Policy Name:  my_media_policy
  _____________________________________________________________
  Type                                      = gateway
  Audio transcoding limit                    = 30
  Audio transrate limit                     = 30
  Video transcoding limit                    = 30
  Inband-dtmf-iw limit                      = 10
  SRTP-iw         limit                     = 20
  Total resource  limit                     = 40
```

Configuring Usage Limits for IPSec Encryption and Decryption and IPSec-Protected Signaling

This task explains how to configure usage limits for IPSec encryption and decryption and IPSec-protected signaling.

**SUMMARY STEPS**

1. configure terminal
2. sbc sbc-name
3. sbe
4. cac-policy-set {policy-set-id | copy {source policy-set-id destination policy-set-id} | swap {source policy-set-id destination policy-set-id} | averaging-period {average-number average-period}}
5. cac-table table-name
6. entry entry-id
7. ipsec maximum registers number
8. ipsec maximum calls number
9. end
10. show sbc sbc-name sbe cac-policy-set policy-set-id detail
11. show sbc sbc-name sbe cac-policy-set policy-set-id table table-name detail
12. show sbc sbc-name sbe cac-policy-set policy-set-id table table-name entry entry-id
## Chapter 7  Implementing Cisco Unified Border Element (SP Edition) Policies

### Limiting Resource Usage

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1**

configure terminal

Example:

Router# configure terminal

| **Step 2**

sbc sbc-name

Example:

Router(config)# sbc mySbc

| **Step 3**

sbe

Example:

Router(config-sbc)# sbe

| **Step 4**

cac-policy-set {policy-set-id | copy {source policy-set-id destination policy-set-id} | swap (source policy-set-id destination policy-set-id) | averaging-period (average-number average-period)}

Example:

Router(config-sbc-sbe)# cac-policy-set 1

| **Step 5**

cac-table table-name

Example:

Router(config-sbc-sbe-cacpolicy)# cac-table t1

| **Step 6**

entry entry-id

Example:

Router(config-sbc-sbe-cacpolicy-cactable)# entry 1

| **Step 7**

ipsec maximum registers number

Example:

Router(config-sbc-sbe-cacpolicy-cactable-entry)# ipsec maximum registers 10

---

### Command or Action Details

1. **configure terminal**
   - Enables the global configuration mode.
   - **Example:**
     
     ```
     Router# configure terminal
     ```

2. **sbc sbc-name**
   - Enters the SBC service mode.
   - **sbc-name**—Name of the SBC.
   - **Example:**
     
     ```
     Router(config)# sbc mySbc
     ```

3. **sbe**
   - Enters the SBE configuration mode.
   - **Example:**
     
     ```
     Router(config-sbc)# sbe
     ```

4. **cac-policy-set {policy-set-id | copy {source policy-set-id destination policy-set-id} | swap (source policy-set-id destination policy-set-id) | averaging-period (average-number average-period)}**
   - Enters the CAC policy set configuration mode within an SBE entity. If the policy set does not exist, it is created.
   - **policy-set-id**—CAC policy set number. The range is from 1 to 2147483647.
   - **Example:**
     
     ```
     Router(config-sbc-sbe)# cac-policy-set 1
     ```

5. **cac-table table-name**
   - Enters the CAC table configuration mode within an SBE policy set. If the CAC table does not exist, it is created.
   - **table-name**—Name of the CAC table.
   - **Example:**
     
     ```
     Router(config-sbc-sbe-cacpolicy)# cac-table t1
     ```

6. **entry entry-id**
   - Enters the mode for configuring an entry in a CAC table. If the entry does not exist, it is created.
   - **entry-id**—ID of the CAC table entry.
   - **Example:**
     
     ```
     Router(config-sbc-sbe-cacpolicy-cactable)# entry 1
     ```

7. **ipsec maximum registers number**
   - Specifies the maximum number of endpoint registrations that can use IPsec encryption and decryption on their signaling link to the SBC.
   - **number**—Number of endpoint registrations. The range is from 1 to 4294967295. The default is 4294967295.
   - **Example:**
     
     ```
     Router(config-sbc-sbe-cacpolicy-cactable-entry)# ipsec maximum registers 10
     ```

---

**Note**

- The keywords and arguments of the **cac-policy-set** command that are not relevant to this section have not been described here. See *Cisco Unified Border Element (SP Edition) Command Reference: Unified Model* for information about these keywords and arguments.

---

**Note**

- This configuration is not used when determining the call scope. In addition, this configuration is not used when the SBC performs the Interconnection Border Control Function (IBCF) because all registrations are stateless and the SBC cannot determine whether a registration is new.
### Command or Action

**Step 8**

```plaintext
ipsec maximum calls number
```

**Example:**

Router(config-sbc-sbe-cacpolicy-cactable-entry)
# ipsec maximum calls 5

**Purpose:** Specifies the maximum number of calls that can use IPsec-protected signaling.
- **number:** Number of calls. The range is from 1 to 4294967295. The default is 4294967295.

**Step 9**

```plaintext
end
```

**Example:**

Router(config-sbc-sbe-cacpolicy-cactable-entry)
# end

**Purpose:** Exits the SBE mode, and returns to the privileged EXEC mode.

**Step 10**

```plaintext
show sbc sbc-name sbe cac-policy-set policy-set-id detail
```

**Example:**

Router(config)# show sbc mySbc sbe cac-policy-set 1 detail

**Purpose:** Shows detailed information pertaining to a CAC policy set.
- **sbc-name:** Name of the SBC.
- **policy-set-id:** ID of the policy set.

**Step 11**

```plaintext
show sbc sbc-name sbe cac-policy-set policy-set-id table table-name detail
```

**Example:**

Router(config)# show sbc mySbc sbe cac-policy-set 1 table t1 detail

**Purpose:** Shows detailed information pertaining to a table in a CAC policy set.
- **sbc-name:** Name of the SBC.
- **policy-set-id:** ID of the policy set.
- **table-name:** Name of the table.

**Step 12**

```plaintext
show sbc sbc-name sbe cac-policy-set policy-set-id table table-name entry entry-id
```

**Example:**

Router(config)# show sbc mySbc sbe cac-policy-set 1 table t1 entry 1

**Purpose:** Shows detailed information pertaining to an entry in a table in a CAC policy set.
- **sbc-name:** Name of the SBC.
- **policy-set-id:** ID of the policy set.
- **table-name:** Name of the table.
- **entry-id:** ID of the CAC table entry.

The following example shows the output of the `show sbc sbe cac-policy-set table entry` command:

```plaintext
Router# show sbc mySbc sbe cac-policy-set 1 table t1 entry 1
SBC Service "mySbc"
CAC Averaging period 1: 60 sec
CAC Averaging period 2: 0 sec

CAC Policy Set 1
  Active policy set: No
  Description:
  First CAC table:
  First CAC scope: global

  Table name: t1
  Description:
  Table type: policy-set
  Total call setup failures (due to non-media limits): 0

  Entry 1
  CAC scope:
  CAC scope prefix length: 0
  Action: Not set
```
Configuration Examples for Implementing Policies

This section provides the following configuration examples:

- Example: Implementing Number Analysis, page 7-156
- Example: Configuring Administrative Domain, page 7-157
- Example: Implementing Call Admission Control Policy Sets and CAC Tables, page 7-159
- Example: Multiple SBC Media Bypass, page 7-161
- Example: Configuring Hunting, page 7-163
- Example: Allowing Asymmetric Payload Types, page 7-164
- Example: Common IP Address Media Bypass, page 7-166
- Example: Limiting Resource Usage, page 7-167
- Example: Configuration the CAC Threshold, page 7-168

Example: Implementing Number Analysis

The following example shows call processing handled with number analysis working with a category routing table in the following manner: 1) shows number analysis, based on number categorization, of a set of dialed digits to determine which is a valid telephone number, 2) shows how the categorized calls are handled with a call routing policy based on category, and 3) shows source address manipulation.

This task configures text address validation and source address manipulation for a number analysis table.

Under 1) for any new call, the SBC inspects the first few digits of the called number that is determined by the “match-prefix” and categorizes the call, based on the category configured under the “na-dst-prefix-table Determine-Category” entry. For example, calls with a prefix of 911 in the destination number are categorized as EMERGENCY calls; calls with a prefix of 919 are Legit_Call, and calls with a prefix of 900 are Blocked_Number calls.

Under 2) routing policy is defined based on category as specified by the “rtg-category-table Category_Routing” table that allows EMERGENCY calls and Legit_Call and rejects all Blocked_Number calls.

call-policy-set 1
  first-inbound-na-table Determine-Category
  first-call-routing-table Category_Routing
  rtg-src-adjacency-table Routing-Table-2
  entry 1
    action complete
    dst-adjacency Adj-502
Example: Configuring Administrative Domain

The following example shows how to configure the administrative domains:

```bash
adjacency sip SIPP1A
admin-domain SIPP1A
    inherit profile preset-access
    signaling-address ipv4 10.10.100.140
    statistics method summary
    signaling-port 7065
    remote-address ipv4 10.10.100.11 255.255.255.255
    signaling-peer 10.10.100.11
    signaling-peer-port 7065
    registration rewrite-register
    attach
```
adjacency sip SIPP1B
admin-domain SIPP1B
inherit profile preset-access
signaling-address ipv4 10.10.100.140
statistics method summary
signaling-port 7066
remote-address ipv4 10.10.100.12 255.255.255.255
signaling-peer 10.10.100.12
signaling-peer-port 7066
registration rewrite-register
attach
adjacency sip Registrar
inherit profile preset-core
signaling-address ipv4 10.10.100.140
statistics method summary
signaling-port 7020 7029
remote-address ipv4 10.10.100.12 255.255.255.255
signaling-peer 10.10.100.12
signaling-peer-port 7068
registration contact username passthrough
registration target address 10.10.100.12
registration target port 7069
attach
cac-policy-set averaging-period 1 120
cac-policy-set averaging-period 2 40
cac-policy-set 10
  first-cac-table TAB1
  first-cac-scope src-adjacency
cac-table TAB1
table-type limit adjacency
  entry 1
    match-value SIPP1A

  .
  .
  .
    action cac-complete
    complete
cac-policy-set 20
  first-cac-table TAB1
cac-table TAB1
table-type policy-set
  entry 1
    max-call-rate-per-scope 600 averaging-period 1
    action cac-complete
    complete
cac-policy-set global 20
call-policy-set 10
  first-call-routing-table RTG_TBL
  first-reg-routing-table REG_TBL
rtg-src-adjacency-table RTG_TBL
  entry 1
    match-adjacency SIPP1A
dst-adjacency SIPP1B
    action complete
rtg-src-adjacency-table REG_TBL
  entry 1
    match-adjacency SIPP1A
dst-adjacency Registrar
    action complete
    complete
call-policy-set 20
  first-call-routing-table RTG_TBL
  first-reg-routing-table REG_TBL
rtg-src-adjacency-table RTG_TBL
Example: Implementing Call Admission Control Policy Sets and CAC Tables

The following example shows how to configure call admission control policy sets and CAC tables:

Router# configure terminal
Router(config)# sbc mysbc
Router(config-sbc)# sbe
Router(config-sbc-sbe)# cac-policy-set 1
Router(config-sbc-sbe-cacpolicy)# first-cac-scope global
Router(config-sbc-sbe-cacpolicy)# first-cac-table STANDARD-LIST-BY-ACCOUNT
Router(config-sbc-sbe-cacpolicy)# cac-table STANDARD-LIST-BY-ACCOUNT
Router(config-sbc-sbe-cacpolicy-cactable)# table-type limit dst-account
Router(config-sbc-sbe-cacpolicy-cactable)# entry 1
Router(config-sbc-sbe-cacpolicy-cactable-entry)# match-value SIP-CUSTOMER-1
Router(config-sbc-sbe-cacpolicy-cactable-entry)# max-num-calls 100
Router(config-sbc-sbe-cacpolicy-cactable-entry)# max-call-rate-per-scope 20
Router(config-sbc-sbe-cacpolicy-cactable-entry)# max-bandwidth 1000000 bps
Router(config-sbc-sbe-cacpolicy-cactable-entry)# callee-privacy never
Router(config-sbc-sbe-cacpolicy-cactable-entry)# action cac-complete
Router(config-sbc-sbe-cacpolicy-cactable-entry)# exit
Router(config-sbc-sbe-cacpolicy-cactable)# entry 2
Router(config-sbc-sbe-cacpolicy-cactable-entry)# match-value SIP-CUSTOMER-2
Router(config-sbc-sbe-cacpolicy-cactable-entry)# max-num-calls 100
Router(config-sbc-sbe-cacpolicy-cactable-entry)# max-call-rate-per-scope 20
Router(config-sbc-sbe-cacpolicy-cactable-entry)# max-bandwidth 1000000 bps
Router(config-sbc-sbe-cacpolicy-cactable-entry)# transcode deny
Router(config-sbc-sbe-cacpolicy-cactable-entry)# action cac-complete
Router(config-sbc-sbe-cacpolicy-cactable-entry)# exit
Router(config-sbc-sbe-cacpolicy-cactable)# complete

The following example limits the total number of concurrent calls per SBC (global limit) to 2000 and the number of concurrent calls per source adjacency to 5. If an adjacency has 5 calls that are active, it is not allowed to make the sixth call even if the total number of active calls on the SBC is less than 2000. Also, if the total number of active calls on the SBC is 2000, an adjacency is not allowed to make a call even if it has no active calls.
Router# configure terminal
Router(config)# sbc mySbc
Router(config-sbc)# sbe
Router(config-sbc-sbe)# cac-policy-set 1
Router(config-sbc-sbe-cacpolicy)# first-cac-scope global
Router(config-sbc-sbe-cacpolicy)# first-cac-table first_policy_table
Router(config-sbc-sbe-cacpolicy)# cac-table first_policy_table
Router(config-sbc-sbe-cacpolicy-cactable)# table-type policy-set
Router(config-sbc-sbe-cacpolicy-cactable)# entry 1
Router(config-sbc-sbe-cacpolicy-cactable-entry)# cac-scope src-adjacency
Router(config-sbc-sbe-cacpolicy-cactable-entry)# action cac-complete
Router(config-sbc-sbe-cacpolicy-cactable-entry)# max-num-calls 5
Router(config-sbc-sbe-cacpolicy-cactable-entry)# exit
Router(config-sbc-sbe-cacpolicy-cactable)# cac-policy-set global 1

The following example limits the number of concurrent calls per subscriber to 5 with no global limit:

Router# configure terminal
Router(config)# sbc mySbc
Router(config-sbc)# sbe
Router(config-sbc-sbe)# cac-policy-set 1
Router(config-sbc-sbe-cacpolicy)# first-cac-table first_policy_table
Router(config-sbc-sbe-cacpolicy)# cac-table first_policy_table
Router(config-sbc-sbe-cacpolicy-cactable)# table-type policy-set
Router(config-sbc-sbe-cacpolicy-cactable)# entry 1
Router(config-sbc-sbe-cacpolicy-cactable-entry)# cac-scope src-adjacency
Router(config-sbc-sbe-cacpolicy-cactable-entry)# action cac-complete
Router(config-sbc-sbe-cacpolicy-cactable-entry)# max-num-calls 5
Router(config-sbc-sbe-cacpolicy-cactable-entry)# exit
Router(config-sbc-sbe-cacpolicy-cactable)# complete
Router(config-sbc-sbe-cacpolicy)# exit
Router(config-sbc-sbe)# cac-policy-set global 1

You could also achieve this with the following configuration:

Router# configure terminal
Router(config)# sbc mySbc
Router(config-sbc)# sbe
Router(config-sbc-sbe)# cac-policy-set 3
Router(config-sbc-sbe-cacpolicy)# first-cac-table 1
Router(config-sbc-sbe-cacpolicy)# first-cac-scope src-adjacency
Router(config-sbc-sbe-cacpolicy)# cac-table 1
Router(config-sbc-sbe-cacpolicy-cactable)# table-type limit all
Router(config-sbc-sbe-cacpolicy-cactable)# entry 1
Router(config-sbc-sbe-cacpolicy-cactable-entry)# max-num-calls 5
Router(config-sbc-sbe-cacpolicy-cactable-entry)# exit
Router(config-sbc-sbe-cacpolicy-cactable-entry)# action cac-complete
Router(config-sbc-sbe-cacpolicy-cactable-entry)# exit
Router(config-sbc-sbe-cacpolicy-cactable)# complete
Router(config-sbc-sbe-cacpolicy)# exit
Router(config-sbc-sbe)# cac-policy-set global 1

Both of the above configurations will limit the number of concurrent calls per subscriber to 5. There is no global limit.
In the following example, if the bandwidth used by an adjacency whose source IP address is 1.1.1.1 is less than 1 Mbps, then the call is admitted. Also adjacencies with a source IP address of 2.2.2.2 that use less than 2 Mbps of bandwidth will have their calls admitted.

Router# configure terminal
Router(config)# sbc mySbc
Router(config-sbc)# sbe
Router(config-sbc-sbe)# cac-policy-set 1
Router(config-sbc-sbe-cacpolicy)# first-cac-scope global
Router(config-sbc-sbe-cacpolicy)# first-cac-table first_policy_table
Router(config-sbc-sbe-cacpolicy)# cac-table first_policy_table
Router(config-sbc-sbe-cacpolicy-cactable)# table-type limit src-adjacency
Router(config-sbc-sbe-cacpolicy-cactable)# entry 1
Router(config-sbc-sbe-cacpolicy-cactable-entry)# match-value 1.1.1.1
Router(config-sbc-sbe-cacpolicy-cactable-entry)# action cac-complete
Router(config-sbc-sbe-cacpolicy-cactable-entry)# max-bandwidth 1 Mbps
Router(config-sbc-sbe-cacpolicy-cactable-entry)# exit
Router(config-sbc-sbe-cacpolicy-cactable)# entry 2
Router(config-sbc-sbe-cacpolicy-cactable-entry)# match-value 2.2.2.2
Router(config-sbc-sbe-cacpolicy-cactable-entry)# action cac-complete
Router(config-sbc-sbe-cacpolicy-cactable-entry)# max-bandwidth 2 Mbps
Router(config-sbc-sbe-cacpolicy-cactable-entry)# exit
Router(config-sbc-sbe-cacpolicy-cactable)# complete
Router(config-sbc-sbe-cacpolicy)# exit
Router(config-sbc-sbe)# cac-policy-set global 1

This example allows 10 calls, 100 updates, a max-in-call-msg-rate and a max-out-call-msg-rate of 5000 msg/min for any source adjacency:

Router# configure terminal
Router(config)# sbc mySbc
Router(config-sbc)# sbe
Router(config-sbc-sbe)# cac-policy-set 1
Router(config-sbc-sbe-cacpolicy)# first-cac-scope sub-category
Router(config-sbc-sbe-cacpolicy)# first-cac-table first_policy_table
Router(config-sbc-sbe-cacpolicy)# cac-table first_policy_table
Router(config-sbc-sbe-cacpolicy-cactable)# table-type policy-set
Router(config-sbc-sbe-cacpolicy-cactable)# entry 1
Router(config-sbc-sbe-cacpolicy-cactable-entry)# cac-scope src-adjacency
Router(config-sbc-sbe-cacpolicy-cactable-entry)# action cac-complete
Router(config-sbc-sbe-cacpolicy-cactable-entry)# max-num-calls 10
Router(config-sbc-sbe-cacpolicy-cactable-entry)# max-updates 100
Router(config-sbc-sbe-cacpolicy-cactable-entry)# max-in-call-msg-rate 5000
Router(config-sbc-sbe-cacpolicy-cactable-entry)# max-out-call-msg-rate 5000
Router(config-sbc-sbe-cacpolicy-cactable-entry)# exit
Router(config-sbc-sbe-cacpolicy-cactable)# complete
Router(config-sbc-sbe-cacpolicy)# exit
Router(config-sbc-sbe)# cac-policy-set global 1

Example: Multiple SBC Media Bypass

The following example shows how to configure a media bypass across two or more SBCs as shown in Figure 7-7, when making calls from endpoint 1 to endpoint 2. In the example, the adjacencies configured on each SBC is named access, for endpoint facing adjacency, and core for proxy facing adjacency. To achieve media bypass for calls from endpoint 2 to endpoint 1, two CAC entries with match-value as access and core must be configured with the same settings in the CAC table.
SBC 1:

sbc SBC1
  sbe
    adjacency sip access
      .
      
      media bypass tag 1 enterprise1
      .
      .
    adjacency sip core
      .
      .
  cac-policy-set 1
    cac-table MyTable
      table-type limit src-adjacency
        entry 1
          .
          
          match-value access
            media bypass type full hairpin
caller media bypass enable
callee media bypass enable
action cac-complete
        entry 2
          session-refresh renegotiation suppress
          .
          .

SBC 2:

sbc SBC2
  sbe
    adjacency sip access
      .
      .
      
      media bypass tag 1 enterprise1
      .
      .
    adjacency sip core
      .
      .
  cac-policy-set 1
    cac-table MyTable
      table-type limit src-adjacency
        entry 1
          .
          
          match-value core
            media bypass type full hairpin
caller media bypass enable
callee media bypass enable
action cac-complete
Chapter 7 Implementing Cisco Unified Border Element (SP Edition) Policies

Configuration Examples for Implementing Policies

The following example shows the output of the `show sbc sbe adjacencies detail` command:

```
Router# show sbc SBC1 sbe adjacencies access detail
SBC Service SBC1
    Adjacency access (SIP)
        Media Bypass Tag List:
            Tag 1:         tag1
            Tag 2:         tag2
        Media Bypass Max Out Data Length:     1024
```

The following example shows the output of the `show sbc sbe cac-policy-set table entry detail` command:

```
Router# show sbc SBC1 sbe cac-policy-set 1 table MyTable entry 1 detail
SBC Service "SBC1"
    CAC Policy Set 1
        Active policy set: No
        Description:
            Averaging period: 60 sec
        First CAC table:
            First CAC scope: global
                Table name: MyTable
                Description:
                    Table type: policy-set
                    Entry 1
                    Action: CAC Complete
                    Media Bypass Type: Full Partial
                    Caller Media Bypass: Enabled
                    Callee Media Bypass: Enabled
```

Example: Configuring Hunting

The following example shows how to hunt for other routes or destination adjacencies in case of a failure in a SIP mode:

```
Router# configure terminal
Router(config)# sbc mySBC
Router(config-sbc)# sbe
Router(config-sbc-sbe)# call-policy-set 1
Router(config-sbc-sbe-rtgpolicy)# first-call-routing-table SAMPLE
Router(config-sbc-sbe-rtgpolicy)# first-reg-routing-table SAMPLE
Router(config-sbc-sbe-rtgpolicy)# rtg-src-adjacency-table SAMPLE
Router(config-sbc-sbe-rtgpolicy-rtgtable)# entry 1
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# action complete
dst-adjacency TA1
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# match-adjacency Hunted
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# exit
Router(config-sbc-sbe-rtgpolicy-rtgtable)# entry 2
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# action complete
dst-adjacency TA2
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# match-adjacency Hunted
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# exit
Router(config-sbc-sbe-rtgpolicy-rtgtable)# entry 3
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# action complete
```
The following example shows how to configure Cisco Unified Border Element (SP Edition) to hunt for other H.323 routes or destination adjacencies in case of a failure:

Router# configure terminal
Router(config)# sbc mySBC
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency h323 adjacency-name
Router(config-sbc-sbe-h323)# hunting-trigger noBandwidth
Router(config-sbc-sbe-h323)# hunting-trigger unreachableDestination
Router(config-sbc-sbe-h323)# hunting-mode altEndps
Router(config-sbc-sbe-h323)# exit

Example: Allowing Asymmetric Payload Types

The following example shows how to configure the SBC to specify support for Asymmetric payload types on the mySBC SBC:

Router# configure terminal
Router(config)# sbc mySBC
Router(config-sbc)# sbe
Router(config-sbc-sbe)# cac-policy-set 1
Router(config-sbc-sbe-cacpolicy)# first-cac-table my_table
Router(config-sbc-sbe-cacpolicy)# cac-table TAB1
Router(config-sbc-sbe-cacpolicy-cactable)# table-type policy-set
Router(config-sbc-sbe-cacpolicy-cactable)# entry 1
Router(config-sbc-sbe-cacpolicy-cactable)# action cac-complete
Router(config-sbc-sbe-cacpolicy-cactable-entry)# payload-type asymmetric allowed
Router(config-sbc-sbe-cacpolicy-cactable-entry)# complete
Router(config-sbc-sbe)# cac-policy-set global 1
Router(config-sbc-sbe)# end
Router#

The following example shows a SIP/SIP call with a single CAC Policy Set table allowing Asymmetric payload types:

**Configuration:**

```
cac-policy-set 1
first-cac-table TAB1
 cac-table TAB1
 table-type policy-set
 entry 1
    payload-type asymmetric allowed
 action cac-complete
 complete
```

```completes```
Call succeeds with the following invite, and 200 messages exchanged:

**Invite Sent:**

2010-01-12 16:28:35  
UDP message sent:

```
INVITE sip:service@2.0.0.5:5078 SIP/2.0
Via: SIP/2.0/UDP 2.0.0.3:5078;branch=z9hG4bK-32567-1-0
From: sipp ;tag=32567SIPpTag091
To: sut
Call-ID: 1-3256702.0.0.3
CSeq: 1 INVITE
Contact: sip:sipp@2.0.0.3:5078
Max-Forwards: 70
Subject: Performance Test
Content-Type: application/sdp
Content-Length: 127

v=0
o=user1 53655765 2353687637 IN IP4 2.0.0.3
s=-
c=IN IP4 2.0.0.3
t=0 0
m=audio 6000 RTP/AVP 18
a=rtpmap:18 G729/8000
```

**200 Received:**

2010-01-12 16:28:35  
UDP message received [485] bytes :

```
SIP/2.0 200 OK
Call-ID: 1-3256702.0.0.3
CSeq: 1 INVITE
From: sipp ;tag=32567SIPpTag091
To: sut ;tag=sip+1+1060000+47e93fd7
Via: SIP/2.0/UDP 2.0.0.3:5078;branch=z9hG4bK-32567-1-0
Server: CISCO-SBC/2.x
Content-Length: 146
Contact:
Content-Type: application/sdp

v=0
o=user1 5338645241744 5338645241744 IN IP4 10.10.20.20
s=-
c=IN IP4 10.10.20.20
t=0 0
m=audio 16384 RTP/AVP 18
a=rtpmap:18 G729/8000
```

The following example shows a SIP/SIP call with a single CAC policy set table disallowing Asymmetric payload types:

**Configuration:**

```
cac-policy-set 1
  first-cac-table TAB1
cac-table TAB1
table-type policy-set
  entry 1
    payload-type asymmetric disallowed
    action cac-complete
```
complete

cac-policy-set global 1

Call fails with the following invite and error messages:

**Invite Sent:**

2010-01-12 16:39:09

UDP message sent:

```
INVITE sip:service@2.0.0.5:5078 SIP/2.0
Via: SIP/2.0/UDP 2.0.0.3:5078;branch=z9hG4bK-32584-1-0
From: sip;p;tag=32584SIPpTag091
To: sut
Call-ID: 1-32584@2.0.0.3
CSeq: 1 INVITE
Contact: sip:sipp@2.0.0.3:5078
Max-Forwards: 70
Subject: Performance Test
Content-Type: application/sdp
Content-Length: 127
```

```
v=0
o=user1 53655765 2353687637 IN IP4 2.0.0.3
s=-
c=IN IP4 2.0.0.3
t=0 0
m=audio 6000 RTP/AVP 18
a=rtpmap:18 G729/8000
```

**Error Message:**

```
-----------------------------------------------
Unexpected UDP message received:

SIP/2.0 400 Bad Request
Call-ID: 1-32584@2.0.0.3
CSeq: 1 INVITE
From: sipp ;tag=32584SIPpTag091
To: sut ;tag=sip+1+10b0000+3621b373
Via: SIP/2.0/UDP 2.0.0.3:5078;branch=z9hG4bK-32584-1-0
Server: CISCO-SBC/2.x
Content-Length: 0
Contact:
```

**Example: Common IP Address Media Bypass**

The following example shows how to configure the Common IP Address Media Bypass feature on the access-side-1 adjacency:

```
Router# configure terminal
Router(config)# sbc mySBC
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency sip access-side-1
Router(config-sbe-sbe-adj-sip)# media bypass auto-nat-tag-gen
Router(config-sbe-sbe-adj-sip)# end
Router# show sbc mySBC sbe adjacencies access-side-1 detail
```

SBC Service "mySBC "
Adacency access-side-1 (SIP)
Status: Detached
Example: Limiting Resource Usage

This section describes examples related to implementing the Limiting Resource Usage feature.

In the following example, the local media gateway is configured to support up to 1000 audio transcoded streams.

Router# configure terminal
Router(config)# sbc mySbc
Router(config-sbc)# sbe
Router(config-sbc-sbe)# media-policy audio_limit1
Router(config-sbc-sbe-media-pol)# type gateway
Router(config-sbc-sbe-media-pol)# transcode audio maximum 1000
Router(config-sbc-sbe-media-pol)# exit
Router(config-sbc-sbe)# media-gateway policy type local
Router(config-sbc-sbe-mg-pol)# media limits audio_limit1

In the following example, the remote media gateway at 192.0.2.26 is configured to support up to 1500 audio transcoded streams.

Router# configure terminal
Router(config)# sbc mySbc
Router(config-sbc)# sbe
Router(config-sbc-sbe)# media-policy audio_limit2
Router(config-sbc-sbe-media-pol)# type gateway
Router(config-sbc-sbe-media-pol)# transcode audio maximum 1500
Router(config-sbc-sbe-media-pol)# exit
Router(config-sbc-sbe)# media-gateway policy type remote ipv4 192.0.2.26 port 2000
Router(config-sbc-sbe-mg-pol)# media limits audio_limit2

In the following example, a default media gateway policy is configured to enable media gateways to support up to 2000 audio transcoded streams. This default media gateway policy is applied on a media gateway (local or remote) when no other media policy is applied on the media gateway.

Router# configure terminal
Router(config)# sbc mySbc
Router(config-sbc)# sbe
Router(config-sbc-sbe)# media-policy audio_limit3
Router(config-sbc-sbe-media-pol)# type gateway
Router(config-sbc-sbe-media-pol)# transcode audio maximum 2000
Router(config-sbc-sbe-media-pol)# exit
Router(config-sbc-sbe)# media-gateway policy type default
Router(config-sbc-sbe-mg-pol)# media limits audio_limit3

In the following example, a CAC policy is configured to restrict all destination numbers other than 911 to at most 5 media streams on which audio transcoding or audio transrating can be performed. Note that the CAC table commands to apply this restriction to all numbers other than 911 have not been included in this sequence of commands.

Router# configure terminal
Router(config)# sbc mySbc
Router(config-sbc)# sbe
Router(config-sbc-sbe)# media-policy media_streams_limit1
Router(config-sbc-sbe-media-pol)# type cac-policy
Example: Configuration the CAC Threshold

The following example shows how to configure a charge of 10 per session and a call admission limit of 50, which allows 5 calls per second (50/10) through the system:

```
Router(config)# call admission new-model
Router(config)# call admission limit 50
Router(config)# call admission pppoe 10 1
```

Configuration Examples for Implementing Call Routing

This section provides the following configuration examples:

- Example: Routing with No Load Balancing, page 7-168
- Example: Least Cost Routing, page 7-169
- Example: Weighted Routing, page 7-170
- Example: Time-Based Routing, page 7-170
- Example: Regular Expression Based Routing, page 7-174
- Example: Trunk-Group ID Routing, page 7-174

Example: Routing with No Load Balancing

```
Router# configure terminal
Router(config)# sbc mySbc
Router(config-sbc)# sbe
Router(config-sbc-sbe)# call-policy-set 1
Router(config-sbc-sbe-rtgpolicy)# first-call-routing-table start_routing
Router(config-sbc-sbe-rtgpolicy)# rtg-dst-address-table start_routing
Router(config-sbc-sbe-rtgpolicy-rtgtable)# entry 1
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# match-address XXX
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# next-table internal_routing
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# exit
Router(config-sbc-sbe-rtgpolicy-rtgtable)# entry 2
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# match-address XXXX
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# next-table external_routing
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# rtg-src-adjacency-table internal_routing
Router(config-sbc-sbe-rtgpolicy-rtgtable)# entry 1
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# match-address sip_to_foo
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# dst-adjacency sip_to_foo
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# exit
Router(config-sbc-sbe-rtgpolicy)# exit
```
Example: Least Cost Routing

The following example configures a routing table that matches on category and then for each entry routes the call to a different least-cost table to choose the adjacency.

```
Router# configure terminal
Router(config)# sbc mySbc
Router(config-sbc)# sbe
Router(config-sbc-sbe)# call-policy-set 1
Router(config-sbc-sbe-rtgpolicy)# rtg-category-table 1
Router(config-sbc-sbe-rtgpolicy-rtgtable)# entry 1
Router(config-sbc-sbe-rtgpolicy-rtgtable)# match-category internal
Router(config-sbc-sbe-rtgpolicy-rtgtable)# action next-table least_int_cost
Router(config-sbc-sbe-rtgpolicy-rtgtable)# exit
Router(config-sbc-sbe-rtgpolicy-rtgtable)# entry 2
Router(config-sbc-sbe-rtgpolicy-rtgtable)# match-category external
Router(config-sbc-sbe-rtgpolicy-rtgtable)# action next-table least_ext_cost
Router(config-sbc-sbe-rtgpolicy-rtgtable)# exit
Router(config-sbc-sbe-rtgpolicy-rtgtable)# exit
Router(config-sbc-sbe-rtgpolicy)# rtg-least-cost-table least_int_cost
Router(config-sbc-sbe-rtgpolicy-rtgtable)# entry 1
Router(config-sbc-sbe-rtgpolicy-rtgtable)# cost 10
Router(config-sbc-sbe-rtgpolicy-rtgtable)# dst-adjacency SipAdj1
Router(config-sbc-sbe-rtgpolicy-rtgtable)# action complete
Router(config-sbc-sbe-rtgpolicy-rtgtable)# exit
Router(config-sbc-sbe-rtgpolicy-rtgtable)# entry 2
Router(config-sbc-sbe-rtgpolicy-rtgtable)# cost 50
Router(config-sbc-sbe-rtgpolicy-rtgtable)# dst-adjacency SipAdj2
Router(config-sbc-sbe-rtgpolicy-rtgtable)# action complete
Router(config-sbc-sbe-rtgpolicy-rtgtable)# exit
Router(config-sbc-sbe-rtgpolicy-rtgtable)# exit
Router(config-sbc-sbe-rtgpolicy)# rtg-least-cost-table least_ext_cost
Router(config-sbc-sbe-rtgpolicy-rtgtable)# entry 1
Router(config-sbc-sbe-rtgpolicy-rtgtable)# cost 50
Router(config-sbc-sbe-rtgpolicy-rtgtable)# dst-adjacency SipAdj3
Router(config-sbc-sbe-rtgpolicy-rtgtable)# action complete
Router(config-sbc-sbe-rtgpolicy-rtgtable)# exit
Router(config-sbc-sbe-rtgpolicy-rtgtable)# entry 2
Router(config-sbc-sbe-rtgpolicy-rtgtable)# cost 100
Router(config-sbc-sbe-rtgpolicy-rtgtable)# dst-adjacency SipAdj4
Router(config-sbc-sbe-rtgpolicy-rtgtable)# action complete
```
Example: Weighted Routing

In the above example, no two entries in one table have the same cost, so the weight parameter is left at the default of 1. If two or more entries with equal cost exist, and are selected for routing, then calls are distributed based on the weight configured (weight being the relative weight of an entry with respect to the lowest weight in the table). For example, if there are three entries of equal cost and weights of entry1, entry2, and entry3 are 1, 2, and 4 respectively, entry2 will route twice the number of calls as entry1, and entry3 will route four times the number of calls as entry1.

In the following example, all calls are routed to entry 1 because it has the lowest cost. However if routing fails, the remaining three entries all have the same cost, so the weight parameters determine which entry is picked. 80% of calls will be routed to SipAdj2 by entry 2, and the remaining 20% will be evenly divided between SipAdj3 and SipAdj4 (weights of entry 3 and entry 4 are left at a default of 1).

Example: Time-Based Routing

The following example shows two entries, one that routes traffic to Adj1 at all times and a second with a higher precedence that routes traffic to Adj2 if the time is between 9 AM and 6 PM on a weekday. When the two time periods overlap, the one with the higher precedence is chosen.

The two times ranges in entry 1 and entry 2 overlap. In this case, a call made between 9 AM to 6 PM on weekdays matches on both the entries but entry 2 is preferred due to its higher precedence.

If multiple ranges are specified as in entry 2, the Cisco Unified Border Element (SP Edition) will match the entry only during the intersection of the ranges. For example, entry 2 matches calls made Monday through Friday between 9 AM to 6 PM. The range is not Monday 9 AM to Friday 6 PM.
Router (config-sbc-sbe-rtgpolicy-rtgtable-entry)# match-time time date yr 2006 2020 mon 1 12 day 1 31
Router (config-sbc-sbe-rtgpolicy-rtgtable-entry)# precedence 5
Router (config-sbc-sbe-rtgpolicy-rtgtable-entry)# dst-adjacency SipAdj1
Router (config-sbc-sbe-rtgpolicy-rtgtable-entry)# action complete
Router (config-sbc-sbe-rtgpolicy-rtgtable-entry)# exit
Router (config-sbc-sbe-rtgpolicy-rtgtable)# entry 2
Router (config-sbc-sbe-rtgpolicy-rtgtable-entry)# match-time time dow 1 5
Router (config-sbc-sbe-rtgpolicy-rtgtable-entry)# match-time time tod hr 9 17 min 0 59
Router (config-sbc-sbe-rtgpolicy-rtgtable-entry)# precedence 10
Router (config-sbc-sbe-rtgpolicy-rtgtable-entry)# dst-adjacency SipAdj2
Router (config-sbc-sbe-rtgpolicy-rtgtable-entry)# action complete

The following example configures a rule that routes traffic through adjacency SipAdj1 at all times, and through SipAdj2 between Monday 9 AM and Friday 6 PM.

Router (config)# sbc mySbc
Router (config-sbc)# sbe
Router (config-sbc-sbe)# call-policy-set 1
Router (config-sbc-sbe-rtgpolicy)# rtg-time-table table1
Router (config-sbc-sbe-rtgpolicy-rtgtable)# entry 1
Router (config-sbc-sbe-rtgpolicy-rtgtable-entry)# match-time time date yr 2006 2020 mon 1 12 day 1 31
Router (config-sbc-sbe-rtgpolicy-rtgtable-entry)# precedence 5
Router (config-sbc-sbe-rtgpolicy-rtgtable-entry)# dst-adjacency SipAdj1
Router (config-sbc-sbe-rtgpolicy-rtgtable-entry)# action complete
Router (config-sbc-sbe-rtgpolicy-rtgtable-entry)# exit
Router (config-sbc-sbe-rtgpolicy-rtgtable)# entry 2
Router (config-sbc-sbe-rtgpolicy-rtgtable-entry)# match-time time dow 1 5
Router (config-sbc-sbe-rtgpolicy-rtgtable-entry)# match-time time tod hr 9 23 min 0 59
Router (config-sbc-sbe-rtgpolicy-rtgtable-entry)# precedence 10
Router (config-sbc-sbe-rtgpolicy-rtgtable-entry)# dst-adjacency SipAdj2
Router (config-sbc-sbe-rtgpolicy-rtgtable-entry)# action complete
Router (config-sbc-sbe-rtgpolicy-rtgtable-entry)# exit
Router (config-sbc-sbe-rtgpolicy-rtgtable)# entry 3
Router (config-sbc-sbe-rtgpolicy-rtgtable-entry)# match-time time dow 2 4
Router (config-sbc-sbe-rtgpolicy-rtgtable-entry)# precedence 10
Router (config-sbc-sbe-rtgpolicy-rtgtable-entry)# dst-adjacency SipAdj2
Router (config-sbc-sbe-rtgpolicy-rtgtable-entry)# action complete
Router (config-sbc-sbe-rtgpolicy-rtgtable)# entry 4
Router (config-sbc-sbe-rtgpolicy-rtgtable-entry)# match-time time dow 5 5
Router (config-sbc-sbe-rtgpolicy-rtgtable-entry)# match-time time tod hr 0 17 min 0 59
Router (config-sbc-sbe-rtgpolicy-rtgtable-entry)# precedence 10
Router (config-sbc-sbe-rtgpolicy-rtgtable-entry)# dst-adjacency SipAdj2
Router (config-sbc-sbe-rtgpolicy-rtgtable-entry)# action complete

In the configuration above, entry 2, entry 3, and entry 4 together specify the range Monday 9:00 AM through Friday 6:00 PM. This could also be accomplished by having one route for the entire time Monday through Friday with separate ranges to divert traffic during nights as follows:

Router (config)# sbc mySbc
Router (config-sbc)# sbe
Router (config-sbc-sbe)# call-policy-set 1
Router (config-sbc-sbe-rtgpolicy)# rtg-time-table table1
Router (config-sbc-sbe-rtgpolicy-rtgtable)# entry 1
Router (config-sbc-sbe-rtgpolicy-rtgtable-entry)# match-time time date yr 2006 2020 mon 1 12 day 1 31
Router (config-sbc-sbe-rtgpolicy-rtgtable-entry)# precedence 5
Router (config-sbc-sbe-rtgpolicy-rtgtable-entry)# dst-adjacency SipAdj1
Router (config-sbc-sbe-rtgpolicy-rtgtable-entry)# action complete
Router (config-sbc-sbe-rtgpolicy-rtgtable-entry)# exit
Router (config-sbc-sbe-rtgpolicy-rtgtable)# entry 2
Router (config-sbc-sbe-rtgpolicy-rtgtable-entry)# match-time time dow 1 5
Router (config-sbc-sbe-rtgpolicy-rtgtable-entry)# precedence 10
The following example shows how to configure a rule that would route traffic through adjacencies SipAdj1 and SipAdj2 on Monday and Wednesday, respectively, between 9 AM and 6 PM, and through SipAdj3 at all other times.

Router# configure terminal
Router(config)# sbc mySbc
Router(config-sbc)# sbe
Router(config-sbc-sbe)# call-policy-set 1
Router(config-sbc-sbe-rtgpolicy)# rtg-time-table table1
Router(config-sbc-sbe-rtgpolicy-rtgtable)# entry 1
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# match-time date yr 2006 2020 mon 1 12 day 1 31
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# precedence 5
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# dst-adjacency SipAdj3
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# action complete
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# entry 2
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# match-time dow 1 1
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# match-time tod hr 9 17 min 0 59
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# precedence 10
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# dst-adjacency SipAdj1
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# action complete
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# entry 3
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# match-time dow 3 3
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# match-time tod hr 9 17 min 0 59
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# precedence 10
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# dst-adjacency SipAdj2
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# action complete

The following example shows how to configure a rule that would route traffic through adjacency SipAdj1 on Saturdays and Sundays between 01 Mar 2008 through 30 Mar 2009, and through SipAdj2 all other times.

Router# configure terminal
Router(config)# sbc mySbc
Router(config-sbc)# sbe
Router(config-sbc-sbe)# call-policy-set 1
Router(config-sbc-sbe-rtgpolicy)# rtg-time-table table1
Router(config-sbc-sbe-rtgpolicy-rtgtable)# entry 1
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# match-time date yr 2006 2020 mon 1 12 day 1 31
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# precedence 5
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# dst-adjacency SipAdj3
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# action complete
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# entry 2
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# match-time dow 1 1
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# match-time tod hr 9 17 min 0 59
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# precedence 10
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# dst-adjacency SipAdj1
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# action complete
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# entry 3
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# match-time dow 6 7
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# match-time tod 0 0 0 23
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# action complete

The following example shows how to configure a rule that would route traffic through adjacencies SipAdj1 and SipAdj2 on Monday and Wednesday, respectively, between 9 AM and 6 PM, and through SipAdj3 at all other times.

Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# dst-adjacency SipAdj2
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# action complete
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# exit
Router(config-sbc-sbe-rtgpolicy-rtgtable)# entry 3
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# match-time dow 1 1
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# match-time tod hr 0 8 min 0 59
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# precedence 20
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# dst-adjacency SipAdj2
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# action complete
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# exit
Router(config-sbc-sbe-rtgpolicy-rtgtable)# entry 4
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# match-time dow 5 5
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# match-time tod hr 18 23 min 0 59
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# precedence 20
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# dst-adjacency SipAdj2
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# action complete

The following example shows how to configure a rule that would route traffic through adjacency SipAdj1 on Saturdays and Sundays between 01 Mar 2008 through 30 Mar 2009, and through SipAdj2 all other times.

Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# dst-adjacency SipAdj2
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# action complete
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# exit
Router(config-sbc-sbe-rtgpolicy-rtgtable)# entry 3
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# match-time dow 1 1
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# match-time tod hr 9 17 min 0 59
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# precedence 10
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# dst-adjacency SipAdj2
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# action complete

The following example shows how to configure a rule that would route traffic through adjacencies SipAdj1 and SipAdj2 on Monday and Wednesday, respectively, between 9 AM and 6 PM, and through SipAdj3 at all other times.
The following example shows how to configure a rule that would route traffic through adjacency SipAdj1 between 10:00 PM and 6:00 AM from Friday to Monday, and through SipAdj2 otherwise.

```
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# precedence 10
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# dst-adjacency SipAdj1
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# action complete
```

Note Time and day of the week are wrapping ranges, so the minimum can be larger than the maximum. For example, a single routing entry with the ranges Friday through Monday and 22:00 through 06:00 will match before 6 AM and after 10 PM on Friday, Saturday, Sunday and Monday.

In the following example, a user has all his routers running GMT no matter where they were so that they can be synchronized. But one router in New York has a time-based routing table that routes traffic to SipAdj1 at all times apart from Monday through Friday from 9 AM to 6 PM when it routes traffic to SipAdj2. The user wants these match times to refer to local time so it is necessary enter a time-offset command (New York is five hours behind GMT) as shown in the example below.

```
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# use-time-offset
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# precedence 5
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# dst-adjacency SipAdj2
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# action complete
```

```
Router(config-sbc-sbe-rtgtable-entry)# match-time tod hr 9 17 min 0 59
Router(config-sbc-sbe-rtgtable-entry)# match-time dow 6 7
```

```
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# use-time-offset
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# precedence 10
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# dst-adjacency SipAdj2
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# action complete
```
Example: Regular Expression Based Routing

The following example shows how to configure the regular expression based routing to match the user name or domain part of a source or destination SIP URI.

Router# configure terminal
Router(config)# sbc mySbc
Router(config-sbc)# sbe
Router(config-sbc-sbe)# call-policy-set 1
Router(config-sbc-sbe-rtgpolicy)# rtg-dst-address-table MyRtgTable
Router(config-sbc-sbe-rtgpolicy-rtgtable)# entry 1
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# match-address user regex
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# exit
Router(config-sbc-sbe-rtgpolicy-rtgtable)# exit
Router(config-sbc-sbe-rtgpolicy)# rtg-src-domain-table MyRtgTable
Router(config-sbc-sbe-rtgpolicy-rtgtable)# entry 1
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# match-domain cisco.com regex
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# exit
Router(config-sbc-sbe-rtgpolicy-rtgtable)# exit
Router(config-sbc-sbe-rtgpolicy)# exit

Example: Trunk-Group ID Routing

The following example shows how to configure the TGID routing to match the TGID parameters of a source or destination SIP URI.

Router# configure terminal
Router(config)# sbc mysbc
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency sip adj1
Router(config-sbc-sbe-adj-sip)# tgid-routing
Router(config-sbc-sbe-adj-sip)# exit
Router(config-sbc-sbe)# call-policy-set 1
Router(config-sbc-sbe-rtgpolicy)# rtg-src-trunk-group-id-table MyRtgTable
Router(config-sbc-sbe-rtgpolicy-rtgtable)# entry 1
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# action complete
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# dst-adjacency SIP-AS540-PSTN-GW2
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# match-type tgid
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# tgid-context example-domain tgid
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# exit
Router(config-sbc-sbe-rtgpolicy-rtgtable)# exit
Call Duration Monitoring

Cisco Unified Border Element (SP Edition) supports the Call Duration Monitoring feature that is used to gracefully terminate calls whose duration has exceeded a configured maximum amount of time. You can configure the maximum call duration to be applied to a call.

Using this feature, the SBC can terminate SIP, H.323, and SIP to H.323 interworked calls, regardless of the signaling and media activity within those calls.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).

For a complete description of the commands used in this chapter, refer to the Cisco Unified Border Element (SP Edition) Command Reference: Unified Model at:


For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

Note

This feature is supported in the unified model for Cisco IOS XE Release 2.5 and later.

Feature History for Call Duration Monitoring Feature

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 2.5</td>
<td>The Call Duration Monitoring feature was introduced on the Cisco IOS XR.</td>
</tr>
</tbody>
</table>

Contents

This chapter contains the following sections:

- Prerequisites, page 8-2
- Information About Call Duration Monitoring, page 8-2
- Configuration Example—Call Duration Monitoring, page 8-7
Prerequisites

The following prerequisite is required to implement this feature:

Before implementing these features, Cisco Unified Border Element (SP Edition) must already be configured.

Information About Call Duration Monitoring

Cisco Unified Border Element (SP Edition) supports the Call Duration Monitoring feature that is used to gracefully terminate calls whose duration has exceeded a maximum amount of time configured with this feature.

If the duration of a call exceeds the configured maximum set by the `max-call-duration` command, Cisco Unified Border Element (SP Edition) gracefully tears down the call.

- For SIP call branches, the SBC sends a SIP BYE to the endpoints.
- For H.323 call branches, the SBC sends a RELEASE COMPLETE message to the endpoints.

If there is a renegotiation in progress when the maximum duration is reached, the SBC attempts to terminate the call as gracefully as possible.

The SBC can terminate SIP, H.323, and SIP to H.323 interworked calls, regardless of the signaling and media activity within those calls.

The SBC will terminate calls under the following conditions:

- Duration has exceeded a maximum amount of time configured by the `max-call-duration` command for this feature.

  **Note** If the `max-call-duration` command is set to the default of zero (0), this results in disabling the Call Duration Monitoring function; and call duration can be determined by other factors, such as no media flow or calls not answered within a specified period of time.

- If no media has flowed on that call for a specified period of time, as configured by the `media-timeout` command.
- Calls that are not answered within a specified amount of time.

Calls are terminated according to whichever timer expires first.

After this feature is configured, the SBC starts a timer to individually monitor each call passing through it. The timer is started:

- For SIP calls, when the call is connected (not on receipt of the message establishing a new call).
- For H.323 calls, when the first SETUP message establishing a new call is received.

Once the timer has been started it cannot be reset.

However, if SBC re-routs a call during call setup, the maximum call duration timer is restarted because the configured maximum duration may have changed (based on the new routing information).
### Configuring Call Duration Monitoring

This task configures Call Duration Monitoring.

**SUMMARY STEPS**

1. `configure terminal`
2. `sbc service-name`
3. `sbe`
4. `cac-policy-set policy-set-id`
5. `cac-table table-name`
6. `table-type { policy-set | limit { list of limit tables }}`
7. `entry entry-id`
8. `cac-scope { list of scope options }
9. `max-call-duration { num }
10. `action [ next-table goto-table-name | cac-complete ]`
11. `exit`
12. `exit`
13. `complete`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>configure terminal</code></td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>sbc service-name</code></td>
<td>Enters the mode of an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Use the <em>service-name</em> argument to define the name of the service.</td>
</tr>
<tr>
<td>Router(config)# sbc mySbc</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>sbe</code></td>
<td>Enters the mode of an SBE entity within an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> <code>cac-policy-set policy-set-id</code></td>
<td>Enters the mode of CAC policy set configuration within an SBE entity, creating a new policy set if necessary.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe)# cac-policy-set 1</td>
<td></td>
</tr>
</tbody>
</table>
### Step 5

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>cac-table table-name</strong></td>
<td>Enters the mode for configuration of an admission control table (creating one if necessary) within the context of an SBE policy set.</td>
</tr>
</tbody>
</table>

**Example:**

Router(config-sbc-sbe-cacpolicy)# cac-table StandardListByAccount
### Step 6

**Command or Action**

```
step-type {policy-set | limit (list of limit tables)}
```

**Example:**

Router(config-sbc-sbe-cacpolicy-cactable)#
table-type policy-set

**Purpose**

Configures the table type of a CAC table within the context of an SBE policy set.

The **list of limit tables** argument controls the syntax of the match-value fields of the entries in the table. Possible available Limit tables are:

- **account**—Compare the name of the account.
- **adj-group**—Compare the name of the adjacency group.
- **adjacency**—Compare the name of the adjacency.
- **all**—No comparison type. All events match this type.
- **call-priority**—Compare with call priority.
- **category**—Compare the number analysis assigned category.
- **dst-account**—Compare the name of the destination account.
- **dst-adj-group**—Compare the name of the destination adjacency group.
- **dst-adjacency**—Compare the name of the destination adjacency.
- **dst-prefix**—Compare the beginning of the dialed digit string.
- **event-type**—Compare with CAC policy event types.
- **src-account**—Compare the name of the source account.
- **src-adj-group**—Compare the name of the source adjacency group.
- **src-adjacency**—Compare the name of the source adjacency.
- **src-prefix**—Compare the beginning of the calling number string.

**Note**

For Limit tables, the event or message or call matches only a single entry.

Features can be enabled or disabled per adjacency group through CAC configuration the same way this is done per individual adjacencies. The adj-group table type matches on either source or destination adjacency group.

When the policy-set keyword is specified, use the **cac-scope** command to configure the scope within each entry at which limits are applied in a CAC Policy Set table.

**Note**

For Policy Set tables, the event or call or message is applied to all entries in this table.
### Information About Call Duration Monitoring

**Step 7**

**Command or Action**

```
entry entry-id
```

**Example:**

```
Router(config-sbc-sbe-cacpolicy-cactable)# entry 1
```

**Purpose**

Enters the CAC table entry configuration mode to create or modify an entry in an admission control table.

**Step 8**

**Command or Action**

```
cac-scope (list of scope options)
```

**Example:**

```
Router(config-sbc-sbe-cacpolicy-cactable-entry)# cac-scope call
```

**Purpose**

Configures the scope within each of the entries at which limits are applied in a policy set table.

- **list of scope options**—Specifies one of the following strings used to match events:
  - `account`—Events that are from the same account.
  - `adjacency`—Events that are from the same adjacency.
  - `adj-group`—Events that are from members of the same adjacency group.
  - `call`—Scope limits are per single call.
  - `category`—Events that have same category.
  - `dst-account`—Events that are sent to the same account.
  - `dst-adj-group`—Events that are sent to the same adjacency group.
  - `dst-adjacency`—Events that are sent to the same adjacency.
  - `dst-number`—Events that have same destination.
  - `global`—Scope limits are global
  - `src-account`—Events that are from the same account.
  - `src-adj-group`—Events that are from the same adjacency group.
  - `src-adjacency`—Events that are from the same adjacency.
  - `src-number`—Events that have the same source number.
  - `sub-category`—The limits specified in this scope apply to all events sent to or received from members of the same subscriber category.
  - `sub-category-pfx`—The limits specified in this scope apply to all events sent to or received from members of the same subscriber category prefix.
  - `subscriber`—The limits specified in this scope apply to all events sent to or received from individual subscribers (a device that is registered with a Registrar server)
The following is a configuration example for the Call Duration Monitoring feature:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# cac-policy-set 1
Router(config-sbc-sbe-cacpolicy)# cac-table 1
Router(config-sbc-sbe-cacpolicy-cactable)# table-type policy-set
Router(config-sbc-sbe-cacpolicy-cactable)# entry 1
Router(config-sbc-sbe-cacpolicy-cactable-entry)# max-call-duration 6000
```

## Command or Action

**Step 9**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>max-call-duration {num}</code></td>
<td>Configures the maximum duration (in seconds) for which a call may exist.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>Router(config-sbc-sbe-cacpolicy-cactable-entry)# max-call-duration 6000</code></td>
</tr>
<tr>
<td></td>
<td>num range is from 0 to 2147483 seconds</td>
</tr>
<tr>
<td></td>
<td>By default, the max-call-duration is 0, which results in disabling the</td>
</tr>
<tr>
<td></td>
<td>Call Duration Monitoring feature.</td>
</tr>
</tbody>
</table>

**Step 10**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>`action [next-table goto-table-name</td>
<td>cac-complete]`</td>
</tr>
<tr>
<td>Example:</td>
<td><code>Router(config-sbc-sbe-cacpolicy-cactable-entry)# action cac-complete</code></td>
</tr>
<tr>
<td></td>
<td>• Identify the next CAC table to process using the <code>next-table</code> keyword and the <code>goto-table-name</code> argument.</td>
</tr>
<tr>
<td></td>
<td>• Stop processing for this scope using the <code>cac-complete</code> keyword.</td>
</tr>
</tbody>
</table>

**Step 11**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>exit</code></td>
<td>Exits from <code>entry</code> to <code>cactable</code> mode.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>Router(config-sbc-sbe-cacpolicy-cactable-entry)# exit</code></td>
</tr>
</tbody>
</table>

**Step 12**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>exit</code></td>
<td>Exits from <code>cactable</code> to <code>cacpolicy</code> mode.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>Router(config-sbc-sbe-cacpolicy-cactable)# exit</code></td>
</tr>
</tbody>
</table>

**Step 13**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>complete</code></td>
<td>Completes the CAC policy set when you have committed the full set.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>Router(config-sbc-sbe-cacpolicy-cactable-entry)# complete</code></td>
</tr>
</tbody>
</table>

## Configuration Example—Call Duration Monitoring

The following is a configuration example for the Call Duration Monitoring feature:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# cac-policy-set 1
Router(config-sbc-sbe-cacpolicy)# cac-table 1
Router(config-sbc-sbe-cacpolicy-cactable)# table-type policy-set
Router(config-sbc-sbe-cacpolicy-cactable)# entry 1
Router(config-sbc-sbe-cacpolicy-cactable-entry)# max-call-duration 6000
```
IP Realm Support

The IP Realm feature is supported on the Cisco Unified Border Element (SP Edition) unified model. This feature allows the grouping of addresses known to a data border element (DBE) into realms and supports a method for the signaling border element (SBE) to specify which realm it requires an address from. IP Realm support enables an IP realm to be configured under an adjacency and to be associated with a media address pool.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).

For a complete description of the commands used in this chapter, refer to the Cisco Unified Border Element (SP Edition) Command Reference: Unified Model at:


For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

Feature History for Support for IP Realm

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 2.5</td>
<td>This feature was introduced on the unified model on the Cisco IOS XR.</td>
</tr>
</tbody>
</table>

Contents

This module contains the following sections:

- Prerequisites, page 9-1
- Information About IP Realm Support on the Unified Model, page 9-2
- Configuring IP Realm Under an Adjacency—Unified Model, page 9-3

Prerequisites

The following prerequisites are required to implement Support for IP Realm:

Before implementing Support for IP Realm, Cisco Unified Border Element (SP Edition) must already be configured.
Information About IP Realm Support on the Unified Model

The IP Realm feature is supported in the unified model by means of the configuration of the IP realm under an adjacency and the IP realm association with a media address or a pool of media addresses. In effect, the adjacency is configured with the realm it belongs to, and the media address or a pool of media addresses is configured to belong to a realm. A call coming in on an adjacency is matched up with a specific media address or media address pool based on the configured realm.

The IP Realm feature adds support to both the SIP and H.323 adjacency configuration to require calls on specific adjacencies to request addresses from a specific realm. For example, when media addresses are to be allocated, the media pools with realm configuration matching the realm configuration on the adjacency are used. If there is no media pool with a matching realm configuration, then pools without any realm tags are used. If there is one or more pools that can be used, the selection criteria is not deterministic.

Each DBE address range may only belong to a single realm. This realm may be changed while the SBC is activated. However the realm change only affects calls set up after the realm change is made and does not affect calls already in existence.

Each adjacency may only select addresses from a single realm. The realm for an adjacency may be changed at any time, but the changed realm only affects new calls.

If there is no address pool with a matching realm the call setup is rejected, resulting in a SIP request failing with error code 503 “Service Unavailable” and an H.323 release complete with a release completion reason of “gatewayResources.”

Media Address Assignment

The user is able to assign a media address or a media address range to a particular realm. If a realm parameter is specified on an incoming adjacency, the SBC selects a media address or an address from a pool that has a matching realm. This allows users to customize their realm matching to implement features, such as wildcarding of realms.

Note

All of the other address range selection criteria must also match, that is, VPN ID, class of service (for port ranges).

If the IP realm configuration is absent under the adjacency, then an address is selected from a pool with any or no realm.

If an IP realm is specified under the adjacency, but there is no address pool with a matching realm, the call setup is rejected, resulting in an error code from the DBE of 510 “Insufficient resources.”

IP Realm Identifier

The IP Realm Identifier is used to indicate to which packet network the media addresses belong. The IP Realm identifier is a string, which may be in a domain name format, for example, “mynet.net” or any other string format. The format of the realm string is up to the user with certain restrictions.

The IP Realm Identifier should be provisioned between the SBE and the DBE. Each of the different IP realms possibly interconnecting with a DBE should have a different identifier.
Realms strings are case-insensitive and are made up of the characters in Table 9-1.

**Table 9-1**  
**IP Realm Identifier String - Allowed Character Set**

<table>
<thead>
<tr>
<th>Allowed Characters</th>
<th>ASCII</th>
<th>Allowed Characters</th>
<th>ASCII</th>
<th>Allowed Characters</th>
<th>ASCII</th>
</tr>
</thead>
<tbody>
<tr>
<td>A - Z</td>
<td>0x41 - 0x5A</td>
<td>&amp;</td>
<td>0x26</td>
<td>?</td>
<td>0x3F</td>
</tr>
<tr>
<td>a - z</td>
<td>0x61 - 0x7A</td>
<td>!</td>
<td>0x21</td>
<td>@</td>
<td>0x40</td>
</tr>
<tr>
<td>0 - 9</td>
<td>0x30 - 0x39</td>
<td>_</td>
<td>0x5F</td>
<td>^</td>
<td>0x5E</td>
</tr>
<tr>
<td>+</td>
<td>0x2B</td>
<td>/</td>
<td>0x2F</td>
<td>'</td>
<td>0x60</td>
</tr>
<tr>
<td>-</td>
<td>0x2D</td>
<td>*</td>
<td>0x27</td>
<td>~</td>
<td>0x7E</td>
</tr>
<tr>
<td>*</td>
<td>0x2A</td>
<td>$</td>
<td>0x24</td>
<td>\</td>
<td>0x5C</td>
</tr>
<tr>
<td>(</td>
<td>0x29</td>
<td>)</td>
<td>0x29</td>
<td>%</td>
<td>0x25</td>
</tr>
<tr>
<td>)</td>
<td>0x7C</td>
<td>.</td>
<td>0x2E</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Configuring IP Realm Under an Adjacency—Unified Model**

To configure an IP Realm under an adjacency in the unified model, you need to perform both of the following tasks:
- Tag the adjacency with the realm it belongs to using the `realm` command.
- Configure the media address or media addresses in a pool to belong to a realm using the `media-address ipv4` or `media-address pool ipv4` command.

**Tagging an Adjacency with a Realm**

In the SBC unified model, the adjacencies need to be tagged with the realm that they belong to. This will enable subsequent calls to use media addresses from that realm.

The following example shows how to tag the SIP adjacency Cisco-gw with the realm cisco.com:

```
Router(config)# sbc mySbc
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency sip Cisco-gw
Router(config-sbc-sbe-adj-sip)# realm cisco.com
```

The following example shows the running configuration after the SIP adjacency Cisco-gw is tagged with the realm cisco.com:

```
Router# show run
adjacency sip Cisco-gw
signaling-address ipv4 200.100.50.8
realm cisco.com
```

**Configuring a Media Address or a Pool of Media Addresses to Belong to a Realm**

In the SBC unified model, you must configure either a media address or the pool of media addresses to be associated with a realm. If the port range is not configured, the SBC selects the default port range.
The following example configures the media address 40.0.0.1 to belong to the cisco.com realm:

Router(config-sbc)# media-address ipv4 40.0.0.1 realm cisco.com
Router(config-sbc-media-address)# port-range 10000 20000 any

The following example configures a pool of media addresses from which the SBC can select. The SBC can select any address from 40.0.0.2 to 40.0.0.31 as the media address to be associated with the cisco.com realm:

Router(config-sbc)# media-address pool ipv4 40.0.0.2 40.0.0.31 realm cisco.com
Router(config-sbc-media-address)# port-range 10000 20000 any

The following example shows the running configuration after configuring media address 40.0.0.1 to the cisco.com realm:

Router# show run
media-address ipv4 40.0.0.1 realm cisco.com
port-range 10000 20000 any

Show Commands—Unified Model

The following are show commands that can be used to display IP realm information in the unified model.

The `show sbc dbes addresses` command lists the H.248 control addresses, media addresses, and IP realm information configured on a DBE:

Router# show sbc global dbes addresses

SBC Service "global"
No controllers configured.
Media-Address: 40.0.0.1
VRF: Global
Port-Range (Service-Class): 10000-20000 (any)
Realm: cisco.com

The `show sbc sbe adjacencies` command lists the adjacencies information, including the IP realm information, configured on an SBE:

Router# show sbc global sbe adjacencies Cisco-gw detail

SBC Service "global"
Adjacency Cisco-gw (SIP)
Status: Detached
Signaling address: 111.45.103.119:default
Signaling-peer: :5060 (Default)
Force next hop: No
Account:
Group: None
In header profile: Default
Out header profile: Default
In method profile: Default
Out method profile: Default
In body profile: None
Out body profile: None
In UA option prof: Default
Out UA option prof: Default
In proxy opt prof: Default
Out proxy opt prof: Default
Priority set name: None
Local-id: None
Rewrite REGISTER: Off
Target address: None
NAT Status: Auto Detect
Reg-min-expiry: 3000 seconds
Fast-register: Enabled
Fast-register-int: 30 seconds
Register aggregate: Disabled
Registration Required: Disabled
Register Out Interval: 0 seconds
Parse username params: Disabled
Supported timer insert: Disabled
Suppress Expires: Disabled
p-asserted-id header-value: not defined
p-assert-id assert: Disabled
Authenticated mode: None
Authenticated realm: None
Auth. nonce life time: 300 seconds
IMS visited NetID: None
Inherit profile: Default
Force next hop: No
Home network Id: None
UnEncrypt key data: None
SIPI passthrough: No
Passthrough headers: Media passthrough: No
Client authentication: No
Incoming 100rel strip: No
Incoming 100rel supp: No
Out 100rel supp add: No
Out 100rel req add: No
Parse TGID parms: No
IP-FQDN inbound: IP-FQDN outbound: FQDN-IP inbound: FQDN-IP outbound:
Outbound Flood Rate: None
Hunting Triggers: Global Triggers
Add transport=tls param: Disabled
Redirect mode: Pass-through
Security: Untrusted-Unencrypted
Ping: Disabled
Ping Interval: 32 seconds
Ping Life Time: 32 seconds
Ping Peer Fail Count: 3
Ping Trap sending: Enabled
Ping Peer Status: Not Tested
Rewrite Request-uri: Disabled
Registration Monitor: Disabled
DTMF SIP NOTIFY Relay: Enabled
DTMF SIP NOTIFY Interval: 2000
DTMF SIP default duration: 200
DTMF Preferred Method: SIP NOTIFY
Realm: cisco.com
Statistics setting: Disabled
Managing Emergency Calls

With the excessive use of VoIP, it is important to understand the various situations that might need prioritized attention. One of the most important situations that must be handled immediately is emergency service numbers, for example, using VoIP to dial a 911 number, which requires immediate attention. It is imperative to provide dedicated and immediate high-priority access for emergency numbers and calls with the specified SIP resource priority header. To analyze and examine the number of emergency calls, the Emergency Call feature has been implemented on the Cisco ASR 1000 Series Router.

Feature History for Managing Emergency Calls

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 2.5</td>
<td>H.323-to-SIP support for emergency calls was introduced on the Cisco ASR1000 Series Aggregation Services Routers.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 3.2S</td>
<td>The Emergency Call Statistics feature was introduced on the Cisco ASR 1000 Series Routers.</td>
</tr>
</tbody>
</table>

Contents

This chapter contains the following sections:

- H.323 to SIP Support for Emergency Calls, page 10-1
- Overview: Emergency Call Statistics, page 10-2
- Restrictions for Emergency Calls, page 10-2
- Performance Impact, page 10-2

H.323 to SIP Support for Emergency Calls

Cisco Unified Border Element (SP Edition) supports H.323 to SIP call routing for emergency calls. Cisco Unified Border Element (SP Edition) routes voice and video calls according to the configured session routing policy. A call is categorized as “emergency” based on the dialed number or on the Resource-Priority header if it is originated on the SIP side. Based on the emergency categorization, special routing and Call Admission Control (CAC) logic is applied.
Overview: Emergency Call Statistics

Certain emergency numbers, such as 911, 999, and so on, need more priority than normal calls. Using the Session Border Controller (SBC), you can configure and define a category and assign a priority for emergency numbers. It is important to know how many such emergency calls are currently in progress, and under which category the calls have been classified.

The Emergency Call feature has been introduced to analyze the number of emergency calls that are assigned to a particular category or a specific priority. The scope of displaying the emergency call statistics varies. The emergency call statistics can either be displayed globally for the SBC system or for a specific adjacency. To display category-wise and priority-wise emergency calls globally for the SBC system, use the `show sbc sbcname sbe call-stats global emergence` command. To display category and priority-wise calls for a specific adjacency, use the `show sbc sbc sbe call-stats adjacency word emergence` command. The per-adjacency statistic calls displays both received and sent calls separately on that adjacency.

The emergency call identification can be performed with the help of different mechanisms:

- Dedicated call setup priority information, such as the SIP resource priority header.
- Analysis of dialled number for known emergency numbers, such as 911, 999, and so on.
- Configuration based on the CAC policy.

Note

The Billing Manager must be enabled and in active state for displaying the emergency call statistics.

Restrictions for Emergency Calls

The following restriction is applicable to the Emergency feature:

- The display of emergency call statistics is always disabled if Billing Manager is not active.

Performance Impact

The Emergency Call feature requires the Billing Manager to be configured and in active state. The performance cost of running the emergency calls feature is 2%. When emergency call statistics are not requested, there is no performance cost, and only the Billing Manager needs to be configured and active. The base occupancy of this feature is less than 0.1%, and the per-call occupancy cost of this feature is less than 0.25%. When emergency call statistics are requested (global or per adjacency), the requested statistics are displayed within half a second.
Unexpected Source Address Alerting

You can configure Cisco Unified Border Element (SP Edition) to provide alerts for any unexpected source addresses that are received. After an unexpected source address is received, a log is created and a Simple Network Management Protocol (SNMP) trap is generated.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).

For a complete description of the commands used in this chapter, refer to the Cisco Unified Border Element (SP Edition) Command Reference: Unified Model at:


To locate documentation for other commands that appear in this chapter, use the command reference master index, or search online.

Note

For Cisco IOS XE Release 2.4, this feature is supported in both the unified model and the distributed model.

Feature History for Unexpected Source Address Alerting

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 2.4</td>
<td>This feature was introduced for the unified model on the Cisco ASR 1000 Series Aggregation Services Routers.</td>
</tr>
</tbody>
</table>

Contents

This module contains the following sections:

- Prerequisites—Implementing Unexpected Source Address Alerting, page 11-2
- Restrictions for Unexpected Source Address Alerting, page 11-2
- Unexpected Source Address Alerting, page 11-2
- Configuring Unexpected Source Address Alerting, page 11-3
- Examples of Configuring Unexpected Source Address Alerting, page 11-4
Prerequisites—Implementing Unexpected Source Address Alerting

The following prerequisite is required to implement the unexpected source address alerting feature:
Before implementing unexpected source address alerting, Cisco Unified Border Element (SP Edition) must already be configured.

Restrictions for Unexpected Source Address Alerting

Review the following restrictions for unexpected source address alerting:

- This configuration option should only be enabled on trusted networks where any single such instance might indicate a threat to network security.
- Alerts on the same flow are rate-limited as are the total number of alerts reported at any one time to ensure management systems are not flooded with reports. There is not a 1-to-1 correspondence between alerts and incorrect packets.
- Diagnosing and resolving the issue of rogue packets is beyond the scope of the Cisco Unified Border Element (SP Edition) function.
- Any and all packets from unexpected sources are dropped.

Unexpected Source Address Alerting

If a packet with unexpected source address/port is received by the data border element (DBE) on a media address, port, or (if applicable) Virtual Routing Forwarding (VRF) used by a current call, then the DBE creates a log and generates an SNMP trap on the appropriate media-flow-stats MIB.

The log (level 63) is output to the console automatically (by default). The log is a member of the MEDIA debug log group. The log includes the local address, port, and VRF where the packets were received and also the source address and port of the received packet.

An alert is generated the first time an unexpected packet is received on a port after the port is opened for a call. If additional unexpected packets are received on the same media port, additional alerts are generated. Any additional alerts are rate-limited. After the call is completed, the media port is assigned to a new call, and the state is reset. A new alert is then generated if any additional unexpected packets are subsequently received.

The SNMP trap that is generated will contain the following fields:

- The address and port where the unexpected packet was received.
- The address and port where the unexpected packet originated.
# Configuring Unexpected Source Address Alerting

**SUMMARY STEPS**

1. `configure`
2. `sbc sbc-name`
3. `sbe`
4. `unexpected-source-alerting`
5. `end`
6. `show sbc sbc-name dbe media-flow-stats vrf vrf-name [ipv4 A.B.C.D [port] port number]`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>configure terminal</code></td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>sbc sbc-name</code></td>
<td>Creates the SBC service on Cisco Unified Border Element (SP Edition) and enters into SBC configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# sbc mysbc</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>sbe</code></td>
<td>Enters the mode of the signaling border element (SBE) function of the SBC.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> <code>unexpected-source-alerting</code></td>
<td>Sets alerting for unexpected source addresses. The <code>no</code> form of this command removes alerting for any unexpected source addresses that are received.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe)# unexpected-source-alerting</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> <code>exit</code></td>
<td>Exits SBE configuration mode and enters SBC configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe)# exit</td>
<td></td>
</tr>
</tbody>
</table>
Examples of Configuring Unexpected Source Address Alerting

This section provides a sample configuration for configuring unexpected source address alerting including an example of the information added to the media flow statistics.

To configure unexpected source address alerting, use the following commands:

```
configure terminal
sbc mysbc
sbe
unexpected-source-alerting
end
```

### Examples of Configuring Unexpected Source Address Alerting

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 6 end</td>
<td>Exits the SBC configuration mode and returns to Privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc)# end</td>
<td></td>
</tr>
<tr>
<td>Step 7 show sbc service-name dbe media-flow-stats vrf vrf-name ipv4 10.1.1.1 port 24000</td>
<td>Displays detailed information about the media flow statistics configured on the DBE.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# show sbc mysbc dbe media-flow-stats vrf vpn3 ipv4 10.1.1.1 port 24000</td>
<td></td>
</tr>
</tbody>
</table>
DoS Prevention and Dynamic Blacklisting

Denial of Service (DoS) prevention and dynamic blacklisting is used by Cisco Unified Border Element (SP Edition) to block malicious endpoints from attacking the network.

Cisco Unified Border Element (SP Edition) monitors signaling traffic and dynamically detects potential attacks without disrupting the rest of the services that it provides. The attacks can then be blocked internally or externally.

DoS attacks are generally performed on Internet services to deny these services to others. They are usually aimed at the provider of the service, and are either purely malicious vandalism or part of an attempt at extortion.

Blacklisting is the process of matching inbound packets based on parameters, such as source IP addresses, and preventing the packets that match those parameters from being processed.

Dynamic blacklists put in place automatically (subject to a set of configurable constraints) by Cisco Unified Border Element (SP Edition) when it detects an attempt to disrupt traffic flowing through it. Dynamic blacklisting does not require management interference. It can occur within milliseconds of the start of an attack and can change and adapt as the attack changes providing immediate network protection.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).


For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

---

Note
For Cisco IOS XR Software Release 3.2S, this feature is supported in the unified model only.

**Feature History for DoS Prevention and Dynamic Blacklisting**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 2.4</td>
<td>This feature was introduced in Cisco IOS XR along with support for the unified model.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 3.2S</td>
<td>Minor, major, and critical alert traps introduced. The policy-rejection event was renamed as cac-policy-rejection, and the routing-failure event was renamed as rtg-policy-rejection. The na-policy-rejection event was introduced.</td>
</tr>
</tbody>
</table>
Chapter 12  DoS Prevention and Dynamic Blacklisting

Contents

This chapter contains the following sections:

- Prerequisites for DoS Prevention and Dynamic Blacklisting, page 12-2
- Restrictions for DoS Prevention and Dynamic Blacklisting, page 12-2
- Information About DoS Prevention and Dynamic Blacklisting, page 12-2
- Overriding Dynamic Blacklisting Default Thresholds, page 12-4
- Dynamic Blacklisting Behavior, page 12-5
- How to Configure Dynamic Blacklisting, page 12-6
- Examples of Configuring, Removing, and Displaying Dynamic Blacklisting, page 12-10

Prerequisites for DoS Prevention and Dynamic Blacklisting

Following are the prerequisites are required for dynamic blacklisting:

- You must already have Cisco Unified Border Element (SP Edition) configured.
- You need to configure blacklisting to override default blacklisting thresholds when the SBE is configured and before you start using Cisco Unified Border Element (SP Edition).

Restrictions for DoS Prevention and Dynamic Blacklisting

The following are restrictions for DoS prevention and dynamic blacklisting:

- Only Session Initiation Protocol (SIP) traffic is analyzed. Attacks over H.323 are not protected. However, an attack over SIP may also result in H.323 traffic being blocked.
- Port specific blacklist configuration is not possible.

Information About DoS Prevention and Dynamic Blacklisting

Cisco Unified Border Element (SP Edition) monitors the following events as “reasons” for initiating DoS detection policies:

- **authentication-failure**—If Cisco Unified Border Element (SP Edition) is locally authenticating the UAs or peers, then any authentication failure will count as one event.
- **bad-address**—This event is generated when an unexpected source sends a packet that reaches Cisco Unified Border Element (SP Edition); the packet will be dropped.
- **rtg-policy-rejection**—This event is generated when traffic fails to find a match in the routing policy. In Cisco IOS XE Release 3.2S, the routing-failure event is renamed as rtg-policy-rejection.
- **endpoint-registration**—This event is generated when an endpoint is registering through Cisco Unified Border Element (SP Edition) and the registration is rejected.
- **corrupt-message**—This event is generated when a signalling message cannot be decoded by the application or contains a protocol exception/violation.
-- **cac-policy-rejection**—This is a complex category because it monitors CAC policy failures, that is, a negative result from the CAC policy. This category includes rate, count, and bandwidth limits, and makes no distinction between them. In Cisco IOS XE Release 3.2S, the policy-rejection event is renamed as cac-policy-rejection.

-- **spam**—Endpoints may send unwanted or spam calls (sometimes called Spam over Internet Telephony (SPIT)). Spam results from too many unexpected signaling messages. Examples of spam include receipt of a SIP response that does not match an earlier sent request, and receipt of excessive retransmissions of a SIP message.

-- **na-policy-rejection**—This event is generated when there are repeated call rejections due to an invalid source number or destination number. This event is considered as a DoS attack.

There are two types of events that would cause blacklisting: low-level and high-level attacks.

- **Low-level attacks**
  An overwhelming volume of traffic sent at line rate to devices that perform a significant amount of processing per packet.

- **High-level attacks**
  Attacks on any bottlenecks within the signaling plane or application layers.

Blacklist enablement is defined as when an event (for example, authentication-failure) that is being monitored, occurs exceeding the number of times configured (trigger-size) within the window (trigger-period), then activate the dynamic access control list for a time period (timeout).

Any given endpoint can have up to three blacklisted events being monitored at a given time on a per-port, per-address, and per-VPN basis. Within the address source type, there is the following order of precedence:

- Limits configured per specific IPv4 address
- Default limits of the parent VRF address space
- Default limits of the global address space (if different from the parent VRF)
- The hard-coded address limits.

The SBC packet filter (SPF) is a new component designed to defend against low-level attacks. The SPF resides with the Media Packet Forwarder (MPF) component on the network processing unit (NPU) and provides low-level DoS prevention for standalone data border element (DBE) and unified SBC deployment scenarios.

A new component is added to the signaling border element (SBE) to detect high-level attacks and create dynamic blacklists based on these attacks. The dynamic blacklist is configured using the command line interface (CLI). It receives events from other SBE components and generates alerts to start or stop the blacklisting of certain messages. Events that might form part of a high-level attack are detected by other SBE components and sent to the SBE Dynamic Blacklisting Component to collect statistics on their rate of occurrence.

### Blacklist Alert Traps

From Cisco IOS Release XE 3.2S, the blacklist settings are configured to implement alert traps. Minor, major, and critical traps are set to be triggered at much lower thresholds values. Blacklist alert traps do not cause any loss of service and not only generate a log message when the threshold is exceeded, but also an SNMP trap, if configured. To enable SNMP SBC blacklist traps, use the `snmp-server enable traps sbc blacklist` command.

These traps can be monitored and modified to detect a DoS attack.
Overriding Dynamic Blacklisting Default Thresholds

Dynamic blacklisting is on by default. Default thresholds are set for Trigger Size, Trigger Period, and Blacklisting Period for each reason. A reason may be an Authentication Failure, Bad Address, Routing Failure, Endpoint Registration, Corrupt Message, Spam, Routing Policy Rejection, or Number Analysis Policy Rejection.

We highly recommend you configure blacklisting to override default thresholds for call setup and registration messages at the time the SBE is configured and before you start using Cisco Unified Border Element (SP Edition). Doing this will ensure that your planned call setup rate or registration message rate does not trigger spam blacklist that will impede traffic flow. It is important to configure the call setup or registration messages thresholds to be above the messages or registration messages per second rate for each SIP-based call in order for traffic to flow through properly. The default values for Trigger Size, Trigger Period, and Blacklisting Period are 40 events per second, or 4 events per 100 milliseconds. This means that traffic over 40 packets per second would trigger blacklisting.

For the following SIP-based call flow, this example describes how to calculate a suitable trigger size threshold for call setup messages per second:

SIP-based call (caller) has:
Send INVITE
Receive 100 Trying
Receive 180 Ringing
Receive 200 OK to confirm Session Establishment
Send ACK to complete Session Establishment
Send BYE
Receive 200 OK

SIP-based call (callee) has:
Send INVITE
Send 100 Trying
Send 180 Ringing
Send 200 OK to confirm Session Establishment
Receive ACK to complete Session Establishment
Receive BYE
Send 200 OK

There are 14 messages or packets for each SIP-based call. If you have a call setup rate of up to 20 calls per second (CPS), then 14 messages x 20 CPS = 280 messages per second. Therefore for a call setup rate of up to 20 CPS, you would configure a trigger size threshold of at least 280 messages per second.

In the following configuration example, you have raised the trigger size to 280 messages or packets per second:

```
blacklist global
  reason spam
  trigger-size 280
  trigger-period 1 seconds
```

Similar to calculating call setup messages per second, the following example describes how to calculate a suitable trigger size threshold for registration messages:

There is one message per registration per second for each SIP-based call. If you have 20 registrations per second, then 1 messages x 20 registrations = 20 messages per second. Therefore for a registration rate of up to 20 registrations per second, you would configure a trigger size threshold of at least 20 messages per second.
Although Dynamic Blacklisting is on by default, you can turn it off by setting the timeout for every reason to zero. However, note that when timeout is set to zero for any unit value, such as milliseconds or seconds, the unit value returned in a show run command displays as "day." You can use the show sbc sbe blacklist configured-limits command to display the default trigger-size, trigger-period and timeout and configured limits. See Examples of Using the show Commands with Blacklisting section on page 12-11 for an example of this command.

Dynamic Blacklisting Behavior

The following is a description of dynamic blacklisting behavior:

- A global rate limit is applied to ensure that the overall load across all sources and destinations does not exceed the CPU capacity (the default limiter 8000 pps/1000 Mbps).
- The hard-coded initial settings for each event type on each IP address are configured by default to hold 4 events for 100 milliseconds. If the configured values are exceeded, the IP address is blacklisted for 10 minutes.
- If you have an explicitly configured limit for a single IP address or port, any trigger and blocking time values defined in that configuration will override the default. Table 12-1 displays where the parameters of the event limits at each scope for a given message can be configured. The limits are different if the message source is on a global address space or VPN.
- Media packets must match a valid entry in the flow table or they are dropped.

Table 12-1 Priority of Event Limit Parameters

<table>
<thead>
<tr>
<th>Scope of Event Limit</th>
<th>Event Limit Parameter Sources (Highest Priority First)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Global Address Space</td>
</tr>
<tr>
<td>Port</td>
<td>1. Explicit limit for this port</td>
</tr>
<tr>
<td></td>
<td>2. Default for this IP address</td>
</tr>
<tr>
<td>Address</td>
<td>1. Explicit limit for this address</td>
</tr>
<tr>
<td></td>
<td>2. Default for global IP addresses</td>
</tr>
<tr>
<td></td>
<td>3. Hard-coded initial settings</td>
</tr>
<tr>
<td>VPN</td>
<td>Explicit limit for the global address space</td>
</tr>
<tr>
<td></td>
<td>2. Limit set for the global address space</td>
</tr>
</tbody>
</table>

- Valid media packets must not exceed bandwidth limits established in call signaling. Non-conferment packets are dropped.
- Signaling packets are rate-limited by the source port in an attempt to halt forceful packet floods early (the default limiter is 1000 pps/100 mpbs).
- Signaling packets that are not destined to a valid local port are dropped.
- Signaling packets are rate-limited by destination port (the default limiter is 4000 pps/500 Mbps).
- Limits can be configured for specific events from the following source(s): a VPN ID, an IP address, or a port at a specific IP address.
• Default limits on event rates may be defined for all source IP addresses on a VPN, and for all ports on a given IP address. The default limits on each IP address are automatically set at the start of the day, but their parameters can be reconfigured. By default, no event limits are configured for ports. Cisco Unified Border Element (SP Edition) monitors events per IP address by default. You can also configure Cisco Unified Border Element (SP Edition) to monitor an entire VPN or a particular port. If any limit in a VPN is then exceeded, the entire VPN is blacklisted. If a limit for a port is exceeded, the port and its IP address are blacklisted.

• Packets are classified as either signaling or media according to the port from where they are sent:
  – Ports below 10,000 are signaling.
  – Ports above 10,000 are media.

• When only a global address space blacklist is defined (no VRF specific blacklist), this will be used to blacklist addresses in all configured VRFs.

• VRF based blacklist limits will override any per source or address-default limits already set. You cannot use per IP address scope to override behavior in VRF space.

• Cisco Unified Border Element (SP Edition) generates an SNMP trap when a blacklist is activated.

How to Configure Dynamic Blacklisting

You can configure dynamic blacklisting as explained in the following sections:

• Configuring Blacklist Parameters for an IP Address, Port, or VPN, page 12-6
• Configuring an End to Blacklisting, page 12-9

Configuring Blacklist Parameters for an IP Address, Port, or VPN

To configure the event limits for a specific source, use the following commands.

Note

You must configure blacklisting to override the default blacklisting thresholds when the SBE is configured, and before you start using Cisco Unified Border Element (SP Edition).

SUMMARY STEPS

1. configure
2. sbc service-name
3. sbe
4. blacklist ipv4 addr
5. description text
6. reason event
7. trigger-size number
8. trigger-period time
9. critical-alert-size number-of-events
10. major-alert-size number-of-events
Chapter 12      DoS Prevention and Dynamic Blacklisting

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td></td>
</tr>
<tr>
<td>configure terminal</td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td></td>
</tr>
<tr>
<td>sbc service-name</td>
<td>Creates the SBC service on Cisco Unified Border Element (SP Edition) and enters the SBC configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config)# sbc mysbc</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td></td>
</tr>
<tr>
<td>sbe</td>
<td>Enters the SBE entity mode within an SBC service.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td></td>
</tr>
<tr>
<td>blacklist ipv4 addr</td>
<td>Enters the blacklist submode for configuring the event limits for a given source.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config)# blacklist ipv4 25.25.25.5</td>
<td></td>
</tr>
</tbody>
</table>

Note: Event limit parameters that are not configured in this submode are configured with the default, as follows:

- port—port-default value for its address.
- IP address—address-default value for the VPN.
- VPN—value for the global address space.
- global address space—no limit.

**Step 5**

description text

Example:

Router(config-sbc-sbe-blacklist)# description NAT of XYZ Corp

Adds a description for the source and its event limits using a readable text string format.

The no form of this command removes the description.

This description is displayed when the show command is used for this source.
### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| 6    | reason event      | Enters the reason submode for configuring a limit for a specific event type on the source. The no form of this command returns the event limit to its default values. An event includes:  
- authentication-failure—Requests that fail authentication.  
- bad-address—Packets from unexpected addresses.  
- rtg-policy-rejection—Requests that fail to be routed by SBC.  
- endpoint-registration—All endpoint registrations.  
- cac-policy-rejection—Requests that are rejected by the CAC policy.  
- corrupt-message—Signaling packets that are too corrupt to be parsed by the relevant protocol.  
- na-policy-rejection—Requests that are rejected by the configured number analysis policy. |
| 7    | trigger-size number | Defines the number of events from the specified source that are allowed before the blacklisting is triggered and all packets are blocked from the source. Range can be 0 to 65535, |
| 8    | trigger-period time | Defines the period of time that events are considered.  
*time* is expressed as *number unit* where *number* is an integer and *unit* is one of: milliseconds, seconds, minutes, hours, or days. Default period of time is between 10 milliseconds and 23 days. |
| 9    | timeout time | Defines the length of time when packets from the source are blocked if the configured limit is exceeded. *time* can have the following values:  
- 0 = the source is not blacklisted  
- never = the blacklisting is permanent  
- *number unit* where *number* is an integer and *unit* is seconds, minutes, hours, or days  
Default period of time is less than 23 days. |
| 10   | critical-alert-size number-of-events | Defines the number of specified events that must occur before the critical alert is triggered. *number-of-events* can have any value ranging from 1 to 65535. |

---

**Example:**

Router(config-sbc-sbe-blacklist)# reason authentication-failure

Router(config-sbc-sbe-blacklist-reason)# trigger-size 5

Router(config-sbc-sbe-blacklist-reason)# trigger-period 20 milliseconds

Router(config-sbc-sbe-blacklist-reason)# timeout 180 seconds

Router(config-sbc-sbe-blacklist-reason)# critical-alert-size 655
## How to Configure Dynamic Blacklisting

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 11</strong> major-alert-size number-of-events</td>
<td>Defines the number of specified events that must occur before the major alert is triggered. <em>number-of-events</em> can have any value ranging from 1 to 65535.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-blacklist-reason)# major-alert-size 300</td>
<td></td>
</tr>
<tr>
<td><strong>Step 12</strong> minor-alert-size number-of-events</td>
<td>Defines the number of specified events that must occur before the minor alert is triggered. <em>number-of-events</em> can have any value ranging from 1 to 65535.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-blacklist-reason)# minor-alert-size 20</td>
<td></td>
</tr>
<tr>
<td><strong>Step 13</strong> end</td>
<td>Exits the reason mode and enters Privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-blacklist-reason)# end</td>
<td></td>
</tr>
<tr>
<td><strong>Step 14</strong> show sbc service-name sbe blacklist configured-limits</td>
<td>Displays detailed information about the explicitly configured limits. Any values not explicitly defined for each source are displayed in brackets.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# show sbc mysbc sbe blacklist global configured-limits</td>
<td></td>
</tr>
<tr>
<td><strong>Step 15</strong> show sbc service-name sbe blacklist source</td>
<td>List the limits that are currently in place for a specific source (in this example, VPN). This includes any defaults or explicitly configured limits. It also includes any defaults of a smaller scope that are configured at this address. Any values that are not explicitly configured are bracketed (these are the values that are inherited from other defaults).</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# show sbc mysbc sbe blacklist vpn3 ipv4 172.19.12.12</td>
<td></td>
</tr>
<tr>
<td><strong>Step 16</strong> show sbc service-name sbe blacklist current-blacklisting</td>
<td>Lists the limits that are causing the source(s) to be blacklisted.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# show sbc mysbc sbe blacklist current-blacklisting</td>
<td></td>
</tr>
</tbody>
</table>

### Configuring an End to Blacklisting

Use the following command to remove the source from the blacklist:

- **clear sbc service-name sbe blacklist source**

For the *service-name* parameter, enter the name of the SBC.

For the *source* parameter enter the name of the blacklist.
Examples of Configuring, Removing, and Displaying Dynamic Blacklisting

This section provides a sample configuration and output for dynamic blacklisting, removing a source from being blacklisted, and also displaying configured limits.

Example of Configuring Dynamic Blacklisting

This blacklist is configured for global address space with one authentication failure from all possible address sources to be captured within a 100 milliseconds window. The ACL created (blacklist) should never timeout.

```
Router(config-sbc-sbe)# blacklist global
Router(config-sbc-sbe-blacklist)# address-default
Router(config-sbc-sbe-blacklist-addr-default)# reason authentication-failure
Router(config-sbc-sbe-blacklist-addr-default)# timeout never
Router(config-sbc-sbe-blacklist-addr-default)# trigger-size 1
Router(config-sbc-sbe-blacklist-addr-default)# trigger-period 100 milliseconds
```

This blacklist is configured for global address space, five packets from unexpected source within a one minute window. The ACL is to time out in 24 hours.

```
Router(config-sbc-sbe)# blacklist global
Router(config-sbc-sbe-blacklist)# ipv4 10.5.1.21
Router(config-sbc-sbe-blacklist-ipv4)# reason bad-address
Router(config-sbc-sbe-blacklist-ipv4)# timeout 1 days
Router(config-sbc-sbe-blacklist-ipv4-reason)# trigger-size 5
Router(config-sbc-sbe-blacklist-ipv4-reason)# trigger-period 1 minutes
```

Example of Removing a Source from the Blacklist

The following example shows the syntax for removing blacklist from Cisco Unified Border Element (SP Edition):

```
Router# clear sbc mysbc sbe blacklist blacklist
Router#
```

Example of Displaying All the Configured Limits

The following example shows the configured limits for various types of blacklisting:

```
Router# show sbc uut105-1 sbe blacklist configured-limits
```

```
SBC Service 'uut105-1'

Blacklist Defaults
====================
Reason            Trigger     Trigger  Blacklisting    Minor    Major  Critical
                Size      Period     Period     Alert     Alert     Alert
Auth-failure              (4)    (100 ms)     (10 mins)  not set  not set   not set
Bad-address               (4)    (100 ms)     (10 mins)  not set  not set   not set
RTG-policy-rejection      (4)    (100 ms)     (10 mins)  not set  not set   not set
Endpoint-registration     (4)    (100 ms)     (10 mins)  not set  not set   not set
CAC-policy-rejection      (4)    (100 ms)     (10 mins)  not set  not set   not set
Corrupt-message           (4)    (100 ms)     (10 mins)  not set  not set   not set
```
Examples of Configuring, Removing, and Displaying Dynamic Blacklisting

The following example shows the command required to list the limits that are currently in place for a specific source (in this example, VPN). This includes any defaults or explicitly configured limits. It also includes any defaults of a smaller scope that are configured at this address. Any values that are not explicitly configured are bracketed (these are the values that are inherited from other defaults).

Router# `show sbc mysbc sbe blacklist vpn3 ipv4 172.19.12.12`

SBC Service "mySbc" SBE dynamic blacklist vpn3 172.19.12.12

vpn3 172.19.12.12

---------------
Reason Trigger Trigger Blacklisting
Size Period Period Alert Alert Alert
Authentication (20) 10 ms (1 hour)
Bad address (20) 10 ms (1 hour)
Routing (20) 10 ms (1 hour)
Registration (5) 100 ms (10 hours)
Policy (20) 10 ms (1 day)
Corrupt 40 10 ms (1 hour)

Default for ports of vpn3 172.19.12.12

---------------
Reason Trigger Trigger Blacklisting
Size Period Period
Authentication 20 1 sec 1 hour
Bad address 20 1 sec 1 hour
Routing 20 1 sec 1 hour
Registration 5 30 sec 10 hours
Policy 20 1 sec 1 day
Corrupt 20 100 ms 1 hour

The following example shows the command required to list the limits that are causing the source(s) to be blacklisted:

Router# `show sbc mysbc sbe blacklist current-blacklisting`

SBC Service "mySbc" SBE dynamic blacklist current members

Global addresses

---------------
Source Source Blacklist Time
Address Port Reason Remaining
--------- -------- -------- ----
**Examples of Configuring, Removing, and Displaying Dynamic Blacklisting**

```
125.125.111.123  All  Authentication  15 mins
125.125.111.253  UDP 85  Registration  10 secs
144.12.12.4      TCP 80  Corruption  Never ends

VRF: vpn3
======
Source  Source  Blacklist  Time
Address  Port  Reason  Remaining
---------  ------  ---------  ---------
132.15.1.2  TCP 285  Registration  112 secs
172.23.22.2  All  Policy  10 hours
```

The following example shows the configured limits:

```
Router# show sbc MySBC sbe blacklist configured-limits

SBC Service "MySBC"

Blacklist Defaults
====================

<table>
<thead>
<tr>
<th>Reason</th>
<th>Trigger</th>
<th>Trigger</th>
<th>Blacklisting</th>
<th>Minor</th>
<th>Major</th>
<th>Critical</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auth-failure</td>
<td>(4)</td>
<td>(100 ms)</td>
<td>(10 mins)</td>
<td>not set</td>
<td>not set</td>
<td>not set</td>
</tr>
<tr>
<td>Bad-address</td>
<td>(4)</td>
<td>(100 ms)</td>
<td>(10 mins)</td>
<td>not set</td>
<td>not set</td>
<td>not set</td>
</tr>
<tr>
<td>RTG-policy-rejection</td>
<td>(4)</td>
<td>(100 ms)</td>
<td>(10 mins)</td>
<td>not set</td>
<td>not set</td>
<td>not set</td>
</tr>
<tr>
<td>Endpoint-registration</td>
<td>(4)</td>
<td>(100 ms)</td>
<td>(10 mins)</td>
<td>not set</td>
<td>not set</td>
<td>not set</td>
</tr>
<tr>
<td>CAC-policy-rejection</td>
<td>(4)</td>
<td>(100 ms)</td>
<td>(10 mins)</td>
<td>not set</td>
<td>not set</td>
<td>not set</td>
</tr>
<tr>
<td>Corrupt-message</td>
<td>(4)</td>
<td>(100 ms)</td>
<td>(10 mins)</td>
<td>not set</td>
<td>not set</td>
<td>not set</td>
</tr>
<tr>
<td>Spam</td>
<td>(30)</td>
<td>(100 ms)</td>
<td>(10 mins)</td>
<td>not set</td>
<td>not set</td>
<td>not set</td>
</tr>
<tr>
<td>NA-policy-rejection</td>
<td>(4)</td>
<td>(100 ms)</td>
<td>(10 mins)</td>
<td>2</td>
<td>not set</td>
<td>not set</td>
</tr>
</tbody>
</table>

VRF:  172.18.53.56
====================

<table>
<thead>
<tr>
<th>Reason</th>
<th>Trigger</th>
<th>Trigger</th>
<th>Blacklisting</th>
<th>Minor</th>
<th>Major</th>
<th>Critical</th>
</tr>
</thead>
<tbody>
<tr>
<td>NA-policy-rejection</td>
<td>(4)</td>
<td>(100 ms)</td>
<td>(10 mins)</td>
<td>2</td>
<td>not set</td>
<td>not set</td>
</tr>
</tbody>
</table>
```

Note: Watch out for the default configurations already in effect. Only the applied configurations are modified.

This example shows current blacklisting:

```
Router# show sbc MySBC sbe blacklist current-blacklisting

SBC Service "MySBC" SBE dynamic blacklist current members

Global addresses
================

| Source  | Source  | Blacklist | Time    
|---------|---------|-----------|---------|
| Address | Port    | Reason    | Remaining
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>10.5.1.31</td>
<td>All</td>
<td>Authentication</td>
<td>Forever</td>
</tr>
</tbody>
</table>
```
Cisco Unified Border Element (SP Edition) enables interworking between in-channel real-time transport protocol (RTP) signaling using the audio/telephone-event MIME type (RFC 2833) to and from out-of-band signaling using the SIP INFO or SIP NOTIFY method.

The Dual Tone Multifrequency (DTMF) Method Interworking and ACCEPT Header Handling feature introduces an adjacency setting that modifies the only auto detection behavior for INFO method.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller, and may be commonly referred to as the session border controller (SBC) in this document.

For a complete description of the commands used in this chapter, refer to the Cisco Unified Border Element (SP Edition) Command Reference: Unified Model at:


For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

### Feature History for Implementing Interworking DTMF on Cisco Unified Border Element (SP Edition)

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 2.1</td>
<td>Interworking DTMF was introduced on the Cisco IOS XR.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 2.4</td>
<td>Introduced support for DTMF Relay Using SIP NOTIFY Messages on the unified model.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 3.1S</td>
<td>The DTMF Method Interworking and ACCEPT Header Handling feature support was added to Cisco ASR 1000 Series Router.</td>
</tr>
</tbody>
</table>

### Contents

This chapter contains the following sections:

- Restrictions, page 13-2
- Prerequisites—Implementing Interworking DTMF, page 13-2
- Information About Interworking DTMF, page 13-2
- Implementing Interworking DTMF, page 13-4
- DTMF Relay Using SIP NOTIFY Messages, page 13-5
- DTMF Method Interworking and ACCEPT Header Handling, page 13-9
Restrictions

The following are restrictions of the Implementing Interworking DTMF feature:

- When the SBC inspects the accept header in the endpoint’s messages, the absence of the accept header means “application/sdp” is supported.
- When audio transcoding is in operation, the SBC does not support sending and receiving RFC 2833 in-band packets to and from the SBC and interworking RFC 2833 packets with out-of-band SIP INFO or SIP NOTIFY Relay messages on the other call leg.
- The SBC does not support the scenario where a caller only supports sending RFC 2833 in-band packets to a callee that supports both RFC 2833 and out-of-band SIP INFO and SIP NOTIFY Relay. In this case, the DTMF digits received out-of-band on the callee side is not able to be translated into RFC 2833 packets on the caller side.
- The SBC does not support configurable outbound RFC 2833 payload type for SIP to SIP calls when the inbound call side does not support RFC 2833.

Prerequisites—Implementing Interworking DTMF

The following prerequisites are required to implement interworking DTMF:

Before implementing interworking DTMF, Cisco Unified Border Element (SP Edition) must already be configured.

Information About Interworking DTMF

Cisco Unified Border Element (SP Edition) automatically selects the best DTMF Interworking technique based on the combined capabilities of the endpoints in a call. See Figure 13-1 for a sample call flow.

The SBC supports the signaling of DTMF using the following modes:

- Media-stream signalling using RTP payload (RFC2833)
- INFO-based DTMF relay (RFC 2976)
- NOTIFY-based DTMF relay

The SBC can interwork between any of these modes using the most performance efficient methods. DTMF interworking for RFC 2833 in-band packets when transcoding is not supported.

Inspection of the arriving INVITE helps determine the caller’s support for DTMF interworking.

To determine whether the caller supports the INFO method, the SBC inspects the Allow header for the INFO method if the Allow header is present. However, the INVITE must also contain an Accept header that contains application/dtmf-relay for the SBC to detect support for DTMF in the INFO method.

Support for the unsolicited NOTIFY method can be determined by the presence of a Call-Info header indicating the NOTIFY method.

An INFO or NOTIFY message is expected to carry a single DTMF tone with an optional duration. If no duration is specified, the default is 250 milliseconds (ms) for an INFO message and 200 ms for a NOTIFY message.
If the SBC determines that either the INFO method or the NOTIFY method for DTMF is supported by the originator, support for both INFO and NOTIFY methods are advertised by the presence of Call-Info and Accept headers on the outbound call. Interworking between these methods is efficient and improves the probability of finding a suitable method for DTMF interworking.

In the case of interworking of DTMF relay using Network Terminating Equipment (NTE) (RFC 2833) and out-of-band DTMF using SIP INFO or SIP NOTIFY, Cisco Unified Border Element (SP Edition) intercepts the NTE packets with DTMF digits and converts them into the appropriate signaling methods through the Route Processor (RP). In the reverse direction, the RP instructs the Cisco Unified Border Element (SP Edition) to inject NTE DTMF packets into an RTP stream.

### DTMF Packet Generation

When the NTE packets are to be inserted in the middle of a stream that is already sending RTP voice packets, then the NTE packets will replace the RTP voice packets in a one-to-one manner so that subsequent voice packets will not need to update their RTP sequence numbers.

### DTMF Packet Detection

To detect DTMF NTE packets, Cisco Unified Border Element (SP Edition) looks at the payload type of every RTP packet and compares it with that of NTE. In case of a match, Cisco Unified Border Element (SP Edition) looks at the event number to determine that it is a DTMF digit. Cisco Unified Border Element (SP Edition) then copies these packets to the RP. Figure 13-1 illustrates this process.
Implementing Interworking DTMF

The following section describes how to configure the default duration of a DTMF event.

Note that Cisco Unified Border Element (SP Edition) may require you to configure header Allow, header Accept, and method INFO as shown below:

```
sbc test
sbe
sip header-profile default
header Allow
header Accept
sip method-profile default
method INFO
```
Configuring Default Duration of a DTMF Event

This task configures the default duration of a DTMF event.

**SUMMARY STEPS**

1. `configure`
2. `sbc sbc-name`
3. `sbe`
4. `dtmf-duration duration`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1**
*configure terminal* | Enables global configuration mode. |
| **Example:**
Router# configure terminal | |
| **Step 2**
sbc *sbc-name* | Creates the SBC service on the SBC called “mysbc” and enters into SBC configuration mode |
| **Example:**
Router(config)# sbc mysbc dbe | Use the `sbc-name` argument to define the name of the service. |
| **Step 3**
sbe | Enters the mode of the signaling border element (SBE) function of the SBC. |
| **Example:**
Router(config-sbc)# sbe | |
| **Step 4**
dtmf-duration *duration* | Configures the default duration of a DTMF event in milliseconds. |
| **Example:**
Router(config-sbc-vdbe)# dtmf-duration 300 | |

**DTMF Relay Using SIP NOTIFY Messages**

In Cisco IOS XE Release 2.4, Cisco Unified Border Element (SP Edition) adds support for DTMF Relay Using SIP NOTIFY Messages. This is an out-of-band procedure for DTMF relay and is sometimes referred to as NOTIFY-based DTMF Relay.

DTMF tones are the tones that are generated when a telephone key is pressed on a touchtone phone. Sometimes the called endpoint needs to hear those tones, such as when you enter digits during the call in response to a menu. However, low-bandwidth codecs can distort the sound. DTMF relay allows that tone information to be reliably passed from one endpoint to the other. By default, SIP uses in-band signaling, sending the DTMF information in the voice stream. If no DTMF relay method is configured, the tones are sent in-band. However, you can configure DTMF relay to use SIP NOTIFY messages for transmitting DTMF tone information.
Cisco Unified Border Element (SP Edition) supports two out-of-band procedures for DTMF relay. One uses SIP INFO methods, and the other uses SIP NOTIFY methods. The SIP INFO method sends DTMF digits in INFO messages. It is always enabled. When a gateway receives an INFO message containing DTMF relay information, it sends the corresponding tone.

SIP NOTIFY DTMF relay is negotiated by including a Call-Info field in the SIP INVITE and response messages, on a per-adjacency basis. This field indicates an ability to use NOTIFY for DTMF tones and the duration of each tone in milliseconds. When a DTMF tone is generated, the caller sends a NOTIFY message to the callee. When the callee receives the NOTIFY, it responds with SIP 200 OK and plays the DTMF tone.

Note
For Cisco IOS XE Release 2.4 and later, this feature is supported in the unified model only.

You can configure a preferred SIP signaling DTMF transport method for endpoints on an adjacency. If Cisco Unified Border Element (SP Edition) has received DTMF information on a call and is sending it to an endpoint on the adjacency, Cisco Unified Border Element (SP Edition) uses a preferred DTMF method to send the information, provided the endpoint supports this method. You can set one of the following DTMF relay methods as the preferred method:

- SIP NOTIFY DTMF Relay (default value)
- SIP INFO DTMF Relay

Use the `dtmf prefer sip [info | notify]` command to configure the preferred relay method.

The default on the Cisco Unified Border Element (SP Edition) is the SIP-NOTIFY relay method. However, Cisco Unified Border Element (SP Edition) uses the RTP-NTE in-band DTMF relay method if the other side does not support SIP-NOTIFY. If no DTMF relay method is configured, the tones are sent in-band.

When SIP NOTIFY relay is enabled on an adjacency, then:

- The SBC accepts in-call, out-of-subscription NOTIFY messages with a DTMF Payload. These messages are not required to contain a Subscription-State header.
- The SBC accepts a Call-Info header in an INVITE message specifying a telephone-event that indicates support for SIP NOTIFY DTMF Relay.
- Configure the NOTIFY interval. You need to configure the maximum interval in milliseconds that the SBC waits between NOTIFY messages for a single DTMF event.
  
  In this case, the SBC has not received an inbound Call-Info header specifying the negotiated duration, so this value is used instead.

  Use the `dtmf sip notify interval` command.

- You can also configure a default duration. This specifies the duration in milliseconds that the SBC advertises on the outbound DTMF transport method if the inbound side of the call does not supply a duration.

  Use the `dtmf sip default duration` command.
Configuring Default Duration of a SIP NOTIFY DTMF Relay Event

This task configures parameters for a SIP NOTIFY DTMF Relay:

**SUMMARY STEPS**

1. configure terminal
2. sbc sbc-name
3. sbe
4. adjacency sip adjacency-name
5. dtmf prefer sip {info | notify}
6. dtmf sip notify interval int_ms
7. dtmf sip default duration dur_ms
8. end
9. show sbc sbc-name sbe adjacencies adjacency-name detail

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc sbc-name</td>
<td>Creates the SBC service on the SBC and enters into SBC configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# sbc mySBC</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Enters the mode of the signaling border element (SBE) function of the SBC.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> adjacency sip adjacency-name</td>
<td>Configures an adjacency on the SBC and enters the mode of an SBE SIP adjacency.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe)# adjacency sip SoftSwitch</td>
<td>Use the adjacency-name argument to define the name of the service.</td>
</tr>
<tr>
<td><strong>Step 5</strong> dtmf prefer sip {info</td>
<td>notify}</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-adj)# dtmf prefer sip notify</td>
<td></td>
</tr>
</tbody>
</table>
DTMF Relay Using SIP NOTIFY Messages

Chapter 13  Implementing Interworking DTMF

Command or Action | Purpose
--- | ---
Step 6 dtmf sip notify interval *int_ms* | (Optional) Configures the maximum interval in milliseconds that the SBC waits between NOTIFY messages for a single DTMF event. *int_ms* is the duration in milliseconds (ms.) The range is 1 to 65535 ms. The default is 2000 ms.

**Example:**
```
Router(config-sbc-sbe-adj)# dtmf sip notify 1000
```

Step 7 dtmf sip default duration *dur_ms* | (Optional) Specifies the duration in milliseconds that the SBC advertises on the outbound DTMF transport method if the inbound side of the call does not supply a duration. *dur_ms* is the duration in milliseconds (ms). The range is 1 to 65535 ms. The default is 200 ms.

**Example:**
```
Router(config-sbc-sbe-adj)# dtmf sip default duration 300
```

Step 8 end | Exits sip adjacency configuration mode and returns to Privileged EXEC mode.

**Example:**
```
Router(config-sbc-sbe-adj)# end
```

Step 9 show sbc sbc-name sbe adjacencies adjacency-name detail | Display all the fields in the specified SIP adjacency, including that SIP NOTIFY relay is enabled, the interval and default duration in milliseconds.

**Example:**
```
Router# show sbc mySBC sbe adjacencies SoftSwitch detail
```

SIP NOTIFY Examples

The following example disables SIP NOTIFY relay for adjacency ADJ2 and configures SIP INFO as the preferred DTMF Relay method:
```
configure terminal
sbc mySbc
sbe
adj sip ADJ2
dtmf disable sip notify
dtmf prefer sip info
dtmf sip default duration 330
```

The following example displays all the fields in the SoftSwitch SIP adjacency, showing that the SIP NOTIFY relay method is enabled, and the interval and default duration in milliseconds:
```
router# show sbc mySBC sbe adjacencies SoftSwitch detail
SBC Service "mySBC"
Adjacency SoftSwitch (SIP)
  Status: Attached
  Signaling address: 100.100.100.100:5060, VRF Admin
  Signaling-peer: 10.10.51.10:5060
  Force next hop: No
  Account: None
  Group: Default
  In header profile: Default
  Out header profile: Default
  In method profile: Default
  Out method profile: Default
  In UA option prof: Default
  Out UA option prof: Default
  In proxy opt prof: Default
```
DTMF Method Interworking and ACCEPT Header Handling

The SBC can be configured to perform the following functions to support the INFO method in any circumstance:

- Automatically detect support for DTMF in INFO (default behavior).
- Does not send DTMF in INFO, and rejects DTMF in INFO, if received.
- Accepts that DTMF in INFO is supported, regardless of the indication in the Accept header.

Auto detection does not detect support for DTMF-based relay in the following events:

- The Allow header contains the INFO method, but does not have the Accept header.
- The Accept header is present, but does not contain the application/dtmf-relay information. Therefore, the SBC can be configured to assume support of DTMF in INFO.
Configuring DTMF Relay in INFO Message

By default, auto detection of support for DTMF-based relay in INFO message occurs, and therefore, no configuration is required. However, to override auto detection so that support for this method is always assumed, not considering the arriving INVITE message.

This section contains information about the following configurations:
- Configuring SBC to Assume Support for INFO-Based DTMF Relay, page 13-10
- Configuring SBC to Disable INFO-Based DTMF Relay, page 13-11

Configuring SBC to Assume Support for INFO-Based DTMF Relay

This task configures parameters to always assume support for INFO-based DTMF relay.

SUMMARY STEPS

1. configure terminal
2. sbc sbc-name
3. sbe
4. adjacency sip adjacency-name
5. dtmf sip info always-supported
6. end
7. show sbc sbc-name sbe adjacencies adjacency-name detail

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td></td>
</tr>
<tr>
<td>configure terminal</td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td></td>
</tr>
<tr>
<td>sbc sbc-name</td>
<td>Creates the SBC service on the SBC and enters into SBC configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# sbc mySBC</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td></td>
</tr>
<tr>
<td>sbe</td>
<td>Enters the mode of the signaling border element (SBE) function of the SBC.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td></td>
</tr>
<tr>
<td>adjacency sip adjacency-name</td>
<td>Configures an adjacency on the SBC and enters the mode of an SBE SIP adjacency.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe)# adjacency sip SoftSwitch</td>
<td>Use the adjacency-name argument to define the name of the service.</td>
</tr>
</tbody>
</table>
Chapter 13      Implementing Interworking DTMF

DTMF Method Interworking and ACCEPT Header Handling

Configuring SBC to Disable INFO-Based DTMF Relay

This task configures parameters to permanently disable support for DTMF-based relay in INFO.

**SUMMARY STEPS**

1. configure terminal
2. sbc sbc-name
3. sbe
4. adjacency sip adjacency-name
5. dtmf disable sip info
6. end
7. show sbc sbc-name sbe adjacencies adjacency-name detail

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 5</strong></td>
<td></td>
</tr>
<tr>
<td>dtmf sip info always-supported</td>
<td>(Optional) Assumes the INFO method as the preferred DTMF transport method for the endpoints on the adjacency. Note Use the no dtmf sip info command to turn on auto detection of DTMF support.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-adj)# dtmf sip info always-supported</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td></td>
</tr>
<tr>
<td>end</td>
<td>Exits the SIP adjacency configuration mode and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-adj)# end</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td></td>
</tr>
<tr>
<td>show sbc sbc-name sbe adjacencies adjacency-name detail</td>
<td>Displays all the fields in the specified SIP adjacency.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# show sbc mySBC sbe adjacencies SoftSwitch detail</td>
<td></td>
</tr>
</tbody>
</table>
### DTMF Relay Using SIP INFO Message Examples

The following example shows how to configure the SBC to always assume support for INFO-based DTMF relay:

```plaintext
configure terminal
sbc mySbc
sbe
    adjacency sip adj1
dtmf sip info always-supported
```

The following example shows how to configure SBC to disable support for DTMF-based relay in INFO permanently:

```plaintext
configure terminal
sbc mySbc
sbe
    adjacency sip adj1
dtmf disable sip info
```

The following example shows the output of the `show sbc sbe adjacencies detail` command. It also shows that the SBC is configured to always assume support for INFO-based DTMF relay:

```plaintext
Router# show sbc asrlk-sbc sbe adjacencies sipp-1 detail
SBC Service 'asrlk-sbc'
    Adjacency sipp-1 (SIP)
    Status: Attached
```
Signaling address: 10.10.100.120:5080
IPsec server port: 0
Signaling-peer: 10.10.100.10:10000
Signaling-peer status: Not Tested
Signaling-peer priority: 2147483647
Signaling-peer switch: always
Peer status: Not Tested
Current peer index: 0
Force next hop: No
Force next hop select: Out-of-dialog
Admin Domain:
Group: None
In header profile: Default
Out header profile: Default
In method profile: Default
Out method profile: Default
Out error profile: Default
In body profile: None
Out body profile: None
In UA option prof: Default
Out UA option prof: Default
In proxy opt prof: Default
Out proxy opt prof: Default
Priority set name: None
Local-id: None
Rewrite REGISTER: Off
Register contact username: Rewrite
Target address: None
NAT Status: Auto Detect
Reg-min-expiry: 3000 seconds
Fast-register: Enabled
Fast-register-int: 30 seconds
SoftSwitch-shield: Disabled
Expires-header: add-not-present
Register aggregate: Disabled
Registration Required: Disabled
Register Out Interval: 0 seconds
Parse username params: Disabled
Supported timer insert: Disabled
Suppress Expires: Disabled
p-asserted-id header-value: not defined
p-assert-id assert: Disabled
Authenticated mode: None
Authenticated realm: None
Auth. nonce life time: 300 seconds
IMS visited NetID: None
Inherit profile: Default
Force next hop: No
Home network Id: None
UnEncrypt key data: None
SIPi passthrough: No
Passthrough headers:
Media passthrough: Yes
Incoming 100rel strip: No
Incoming 100rel supp: No
Out 100rel supp add: No
Out 100rel req add: No
Parse TGID parms: No
IP-PQDN inbound:
IP-PQDN outbound:
FQDN-IP inbound:
FQDN-IP outbound:
Outbound Flood Rate: None
Hunting Triggers: Global Triggers
Add transport=tls param: Disabled
Redirect mode: Pass-through
Security: Untrusted-Unencrypted
Privacy: Inherit-profile (default)
TLS mutual authentication: No
Ping: Disabled
Ping Interval: 32 seconds
Ping Life Time: 32 seconds
Ping Peer Fail Count: 3
Ping Trap sending: Enabled
Ping Suppression:
Ping Bad Response Code: 300-399
Ping Peer Status: Not Tested
Rewrite Request-uri: Disabled
Registration Monitor: Disabled
DTMF SIP INFO Relay: Always supported
DTMF SIP NOTIFY Relay: Enabled
DTMF SIP NOTIFY Interval: 2000
DTMF SIP default duration: 200
DTMF Preferred Method: SIP NOTIFY
Realm: None
Statistics setting: Summary
IMS RX: Disabled
IMS RFC: Enabled
IMS Nass: Disabled
IMS realm name:
PANI:
Warrant Match-Order: None
Media Bypass Max Out Data Length: 1000
Media Bypass Tag List: None
This chapter describes high availability support for Cisco Unified Border Element (SP Edition) on the Cisco ASR 1000 Series Aggregation Services Routers.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).

For Cisco IOS XE Release 2.4 and later, this feature is supported in both the unified model and the distributed model.

### Feature History for Cisco Unified Border Element (SP Edition) Redundancy—High Availability Support

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 2.4</td>
<td>Added support for Cisco Unified Border Element (SP Edition) unified model.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 2.1</td>
<td>This feature was introduced on the Cisco IOS XR on the data border element (DBE) for the distributed model.</td>
</tr>
</tbody>
</table>

## Contents

This chapter contains the following sections:

- Integrated Session Border Controller High Availability, page 14-2
- Hardware Redundancy, page 14-2
- Software Redundancy, page 14-2
- Route Processor Redundancy (RPR), page 14-3
- SSO Support, page 14-3
- ISSU Support, page 14-4
Integrated Session Border Controller High Availability

The Cisco ASR 1000 Series Routers include the Cisco ASR 1002, Cisco ASR 1004, and Cisco ASR 1006 Routers. The different models support different types of redundancy. Integrated Session Border Controller supports the redundancy available on each model.

On the Cisco ASR 1002 and Cisco ASR 1004 Routers, only software redundancy is available. These routers have dual Cisco IOS software modules running on the same Route Processor, with one active and the other in standby mode. However, these routers can have hardware redundancy by using interchassis hardware redundancy.

The Cisco ASR 1006 Routers offer dual hardware redundancy and software redundancy. Cisco Unified Border Element (SP Edition) high availability is provided in the standard image for the Cisco ASR 1000 Series Routers. There is no special configuration required.

For additional information, see the “High Availability Overview” section in the Cisco ASR 1000 Series Aggregation Services Routers Software Configuration Guide. Also see the Cisco IOS High Availability Configuration Guide for information on high availability features that are on other Cisco platforms and that work identically on the Cisco ASR 1000 Series Aggregation Services Routers.

Hardware Redundancy

Integrated Session Border Controller supports use of a redundant or standby Route Processor (RP) and redundant Embedded Services Processor (ESP) on the Cisco ASR 1006 Router. The Cisco ASR 1006 Router has an ESP as well as an RP for dual hardware redundancy. If the active RP or active ESP hardware fails, the system performs a switchover to the standby RP or standby ESP. RP and ESP hardware redundancy support is independent. An RP failure does not require a switchover of the ESP hardware and an ESP failure does not require an RP switchover.

Hardware redundancy is available only on the Cisco ASR 1006 Router.

Software Redundancy

On the Cisco ASR 1000 Series Routers, Cisco IOS runs as one of many processes within the Cisco IOS XE operating system. This architecture is different than on traditional Cisco IOS, where all processes are run within Cisco IOS. The Cisco ASR 1000 Series Router architecture allows for software redundancy opportunities not available on other Cisco IOS platforms.

Integrated Session Border Controller supports software redundancy by running a Standby peer SBC module within the Standby IOS process. If the Active SBC module fails, then the Active IOS process switches over to the Standby IOS process and the old Standby Integrated SBC module resumes processing as the Active. The Standby IOS process may reside on the same Route Processor as the active IOS process (Cisco ASR 1002 and Cisco ASR 1004 Routers) or it may be on a redundant, standby RP (Cisco ASR 1006 Router).

On the Cisco ASR 1002 and Cisco ASR 1004 Routers, a standby Cisco IOS process is running on the same Route Processor as the active Cisco IOS process. In the event of a Cisco IOS failure, the Router switches to the standby Cisco IOS process. No redundant Route Processor or redundant ESP is available on the Cisco ASR 1002 Series and Cisco ASR 1004 Series Routers.

On the Cisco ASR 1006 Routers, both unified and distributed configurations can operate with a redundant Route Processor and a redundant ESP. In the event of failure of the active Cisco IOS process, the router switches to the standby Cisco IOS process, running on a separate standby Route Processor.
Cisco Unified Border Element (SP Edition) redundancy at the ESP level is provided only if a standby, redundant ESP is used. SBC components running on the active ESP have identical peer components running on the standby ESP. In this case, if the SBC components running on the active ESP fail, then a switchover to the backup ESP occurs.

The following types of software redundancy are supported on Cisco Unified Border Element (SP Edition):

- Route Processor Redundancy (RPR)
- Stateful Switchover (SSO)
- In-Service Software Upgrade (ISSU)

**Route Processor Redundancy (RPR)**

RPR allows you to run with a standby RP or standby Cisco IOS process without state synchronization. In the event of a fatal error on the active RP (or active Cisco IOS process), the system switches to the standby RP (or standby Cisco IOS process), which then completes its initialization. Because all the state information held by the former “Active” is lost, the new “Active” has to configure itself and relearn all the state information.

Upon an RPR-based RP switchover event, all SBC calls already established (in a steady state) at the time of the switchover are lost. SBC calls in the process of being established at the time of the switchover are dropped as gracefully as possible. No new calls can be established briefly after the initial switchover event.

RPR redundancy can allow for Cisco IOS fast software upgrades when ISSU is unavailable. In RPR mode, no Cisco IOS SBC state information is synchronized to the “Standby.” Therefore, all calls are dropped upon an RPR-based switchover.

---

**Note**

RPR is supported on the Cisco ASR 1000 Series Routers while RPR+ is not. You can use Stateful Switchover (SSO) instead of RPR+.

---

**SSO Support**

Integrated Session Border Controller support for Stateful Switchover (SSO) allows for stateful Cisco IOS process switchovers where critical state information is synchronized between one Route Processor used as the active processor and the other RP used as the standby processor, or between active and standby Cisco IOS processes on the same RP. When Cisco IOS is configured for SSO, the SBC module running on the active IOS process constantly “replicates” its internal state to its standby peer SBC module on the standby IOS process. In this way, the standby SBC module is kept in sync with the active IOS process and has all the state information necessary to retain active calls and resume call processing in the event the active IOS process fails and an SSO occurs.

For information on SSO, see the *Cisco IOS High Availability Configuration Guide* at the following URL: http://www.cisco.com/en/US/docs/ios-xml/ios/ha/configuration/12-2sr/ha-12-2sr-book.html
ISSU Support

Integrated Session Border Controller supports In-Service Software Upgrade (ISSU) with a redundant RP or redundant IOS process. The ISSU process allows software to be updated or otherwise modified on a standby RP or standby IOS process while packet forwarding on the active RP or active IOS process continues. For the Cisco ASR 1000 Series Routers, ISSU compatibility depends on the software package being upgraded and the hardware configuration.

Although ISSU between a distributed-only version and a unified version of Cisco IOS XE software may be supported, the unified features introduced in Cisco IOS XE Release 2.4 are not available to the distributed-only version if you should do a software downgrade. In such cases, we advise you to unconfigure unified Cisco Unified Border Element (SP Edition) before performing a downgrade to a Cisco IOS XE software version that does not support unified Cisco Unified Border Element (SP Edition). The same restriction does not apply to a distributed-only Cisco Unified Border Element (SP Edition) configuration.

See the “High Availability Overview” section in the Cisco ASR 1000 Aggregation Services Router Software Configuration Guide for more, updated information on ISSU compatibility.

Interchassis High Availability

The Interchassis High Availability feature provides geographically dispersed multibox redundancy. The unified session border controller (SBC) and the distributed SBC support the box-to-box high availability.

Interchassis High Availability feature is supported by the Cisco ASR 1001 Series Routers, Cisco ASR 1002 Series Routers, Cisco ASR 1004 Series Routers, Cisco ASR 1006 Series Routers, and Cisco ASR 1013 Series Routers.

Cisco Unified Border Element (SP Edition) was earlier known as Integrated Session Border Controller. It is referred to in this document as the session border controller (SBC).

For a complete description of the commands used in this chapter, refer to the Cisco Unified Border Element (SP Edition) Command Reference: Unified Model at:

For information about all the Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

Feature History for Interchassis High Availability

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 3.2S</td>
<td>This feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 3.3S</td>
<td>Added support information pertaining to the Cisco ASR 1006 Series Router, Cisco ASR 1013 Series Router, and interchassis-intrachassis conversion.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 3.7S</td>
<td>Added information about upgrading interchassis redundancy.</td>
</tr>
</tbody>
</table>

Contents

This module contains the following sections:

- Prerequisites for Interchassis High Availability, page 15-2
- Restrictions for Interchassis High Availability, page 15-2
- Information About Interchassis High Availability, page 15-3
- Assigning a Redundancy Group to the SBC, page 15-12
- Managing and Monitoring Interchassis High Availability, page 15-14
Prerequisites for Interchassis High Availability

Following are the prerequisites pertaining to the Interchassis High Availability feature:

- The interfaces shared by the SBC must have the same redundant interface identifier (RII).
- The active device and the standby device must have the same peripheral configuration as the SBC features such as the SBC interfaces, virtual routing and forwarding (VRF), routes, sbc redundancy groups, and so on. The SBC-specific configuration will be replicated to the standby. Therefore, only the active Cisco ASR 1000 Series Router requires the full SBC-specific configuration.
- The active device and the standby device must run on the identical version of the Cisco IOS XE software.
- The active device and the standby device must be connected through an L2 connection for the control path.
- The Embedded Service Processor must be the same on both the active and standby devices. RP's must also match and have similar physical port adapter configuration.
- Network Time Protocol (NTP) must be configured or the clock must be set identical on both Cisco ASR 1000 Series Routers to allow timestamps and call timers to match.
- The latency times must be minimal on all control and data links to prevent timeouts.
- Physically redundant links, such as Gigabit Ether Channel must be used for control and data path.

Restrictions for Interchassis High Availability

Following are the restrictions pertaining to the Interchassis High Availability feature:

- Clustering of more than two SBCs for redundancy is not supported.
- The failover time for a box-to-box application is higher for a non-box-to-box application.
- LAN and MESH scenarios are not supported.
- If a dual IOS daemon is configured, the device does not support the interchassis high availability configuration.
- Only the SBC Active-Standby mode is supported.
- The SBC interfaces must be used for signaling and media addresses. Physical interface IP addresses must not be used.
- VRF’s must be defined in the same order on both active and standby routers for an accurate synchronization of the SBC data.
- When the configuration is replicated to the standby router, it is not committed to the startup configuration, it is in the running configuration. The user must execute the write memory command to commit changes on the standby router that have been synchronized from the active router.
- Coexistence of interchassis high availability and intrachassis high availability is not supported.
- In Cisco ASR 1001 Series Routers, Cisco ASR 1002 Series Routers, and Cisco ASR 1004 Series Routers, the interchassis redundancy is not supported with software redundancy.
In Cisco ASR 1006 Series Routers and Cisco ASR 1013 Series Routers, interchassis redundancy is not supported with intrachassis redundancy. It is supported with a single RP and ESP in the chassis.

When CUBE-SP is in inter-chassis redundancy mode, the customer needs to use the `sync` command in the active box to sync the configuration file from the active box to the standby box so that the latest configuration of CUBE-SP will be synchronized in the running configuration file in the standby box.

Information About Interchassis High Availability

The Interchassis High Availability feature enables the configuration of pairs of routers to act as backups for each other. This feature can be configured to determine the active router based on a number of failover conditions. When a failover occurs, the standby router seamlessly takes over and starts processing call signaling and performing media forwarding.

Groups of redundant interfaces are known as redundancy groups. Figure 15-1 depicts the active-standby device scenario. It shows how the redundancy group is configured for a pair of routers that have a single outgoing interface.

The routers are joined by a configurable control link and data synchronization link. The control link is used to communicate the status of the routers. The data synchronization link is used to transfer stateful information from the SBC, and to synchronize the stateful database for the calls and media flows. Each pair of redundant interfaces are configured with the same unique ID number, also known as the RII.

![Redundancy Group Configuration](image-url)
The status of the redundancy group members is determined through the use of Hello messages sent over the control link. If either of the routers do not respond to a Hello message within the configured amount of time, it is considered that a failure has occurred, and a switchover is initiated. To detect a failure in milliseconds, the control links run the failover protocol integrated with the Bidirectional Forwarding Detection (BFD) protocol. You can configure the following parameters for the Hello messages:

- Active timer
- Standby timer
- Hellotime—The interval at which Hello messages are sent.
- Holdtime—The amount of time before the active or the standby router is declared to be down.

The hellotime defaults to 3 seconds to align with the Hot Standby Router Protocol (HSRP), and the holdtime defaults to 10 seconds. You can also configure these timers in milliseconds by using the `timers hellotime msec` command.

**Note**

B2B HA redundancy holdtime should be at least 3 seconds, and is recommended holdtime is 5 seconds.

**Note**

If you allocate a large amount of memory, for example, 1 GB, to the log buffer, the CPU utilization and memory utilization of the router increases. This issue is compounded if you set small intervals for the hellotime and the holdtime. If you want to allocate a large amount of memory to the log buffer, we recommend that you accept the default values for the hellotime and holdtime. For the same reason, we also recommend that you do not use the `preempt` command.

To determine which pairs of interfaces are affected by the switchover, you must configure the RII for each pair of redundant interfaces.

Priority can be configured in the startup or running configuration, whereas the run-time priority is the priority of the router at any given time. The run-time priority can be similar to the configured priority if no decrements have been made, or it may be lowered based on the interface faults and decrements. The following priority factors can cause a switchover:

- The router with the highest priority value is the active router. If a fault occurs on either the active router or the standby router, the priority of the router is decremented by a configurable amount known as the decrement value. If the priority of the active router falls below the priority of the standby router, a switchover occurs, and the standby router becomes the active router. This default behavior can be overridden by disabling the preemption attribute for the redundancy group. You can also configure each interface to decrease the priority when the L1 state of the interface goes down. This amount overrides the default amount configured for the redundancy group.

**Note**

By default, preemption is not enabled. It can be enabled using the `preempt` command. When preemption is configured, the standby router initiates the failover. However, if you configure SBC on the router, we recommend that you do not use the `preempt` command. If the `preempt` command has been configured and if a failover occurs, the B2B state changes might not progress in a manner that permits a guaranteed amount of time for SBC synchronization.

Each failure event that causes a modification of a redundancy group’s priority generates a syslog entry that contains a time stamp, information about the redundancy group that was affected, the previous priority, the new priority, and a description of the failure event cause.

- When the priority of a router or interface falls below a configurable threshold level, the active router initiates the failover.
A switchover to the standby router can also occur under the following circumstances:

- Power loss or reload occurs on the active router, including crashes.
- The redundancy group on the active router is reloaded manually using the `redundancy application reload group rg-number self` command.

Two consecutive Hello messages that are missed on any monitored interface forces the interface into testing mode. When this occurs, both the units first verify the link status on the interface, and then execute the following tests:

- Network activity test
- ARP test
- Broadcast ping test

### Exclusive Virtual IP and Exclusive Virtual MAC

Virtual IP (VIP) and Virtual MAC are used by the SBC application to control the interfaces that receive traffic. An interface on one device is paired with another interface on another device, and both the interfaces are associated with the same redundancy group. The interface that is associated with an active redundancy group exclusively owns the VIP Address and the Virtual MAC. The Address Resolution Protocol (ARP) process on that device sends ARP replies for ARP requests, if any, pertaining to the VIP, and the Ethernet controller for the interface is programmed to receive the packets destined for the Virtual MAC. When a redundancy group failover occurs, the ownership of the VIP and Virtual MAC changes. The interface associated with the newly active redundancy group sends a gratuitous ARP, and programs the interface's Ethernet controller to accept the packets destined for the Virtual MAC.

### LAN-LAN Topology

The Interchassis High Availability feature supports the LAN-LAN topology. Figure 15-2 shows a LAN-LAN topology. Traffic is often directed to the SBC by configuring static routing in the upstream or downstream routers to an appropriate SBC interface IP address. In addition, the Cisco ASR 1000 Series Routers can participate in dynamic routing with either upstream or downstream routers. The dynamic routing configuration supported on the LAN-facing interfaces can introduce a dependency on routing protocol convergence, thus increasing the failover time.
The Interchassis High Availability feature supports the WAN-LAN topology.

Note
However, asymmetric routing is not supported in the Interchassis High Availability feature.

Figure 15-3 shows a WAN-LAN topology in which the LAN is similar to that present in the LAN-LAN topology. For the WAN, VIP is not required. The SBC interface network can be distributed on both the Cisco ASR 1000 Series Routers through dynamic routing. Routing protocols, such as OSPF, ISIS, and BGP, can run over the WAN links.

For a traffic failover caused by a WAN-facing router failure, the immediate WAN link or other WAN connectivity is dependent on the routing protocol convergence. Although subsecond failover cannot be achieved in these failure scenarios, fault detection can be minimized by tuning the routing protocol keep-alive timers and using the BFD feature, if available. Using the IOS Track feature and decreasing the redundancy group’s priority values on the Cisco ASR 1000 Series Router to trigger failovers when the WAN links have failed, helps minimize the SBC downtime by failing over to a standby router with full connectivity.
Transport by Redundancy Group and SBC

Redundancy group requires each client to establish a connection between the standby and active devices. The Cisco ASR 1000 Series Router platform implementation for box-to-box uses a connection between the standby and active routers using Stream Control Transmission Protocol (SCTP). This connection is used by the Redundancy Facility client to exchange the events and status used to keep the two boxes in synchronization. The platform also has an MCP client that uses a reliable User Datagram Protocol (UDP) connection for exchanging the platform-specific status and events.

The SBC has its own client, and uses a TCP connection for exchanging status, events, and replication data. These connections can be viewed using the `show redundancy application transport clients` command, and the details of the connections, ports, and IP addresses, can be viewed using the `show redundancy application group` command.

Interchassis-Intrachassis Conversion

From Cisco IOS XE Release 3.3S, Interchassis High Availability feature is also supported on Cisco ASR 1006 Series Routers and Cisco ASR 1013 Series Routers.
Intrachassis high availability occurs when the Cisco ASR 1000 Series Router has two routing processors (RP), with one RP in active mode and the other RP in standby mode. Interchassis high availability occurs when there are two Cisco ASR 1000 Series Router, with one router in active mode and the other in standby mode, and each router has one RP.

The following sections list the steps involved in high availability interchassis-intrachassis conversion:

- Intrachassis to Interchassis Conversion, page 15-8
- Interchassis to Intrachassis Conversion, page 15-10

**Intrachassis to Interchassis Conversion**

The following steps describe the procedure involved in dual RPs to single RP box-to-box conversion:

**Step 1** Configure the Cisco ASR 1006 Series Router and Cisco ASR 1013 Series Router with dual RPs and dual forwarding processors (FPs) in the Stateful Switchover (SSO) mode.

**Step 2** Configure the SBC functionality and generate test calls to ensure proper operation.

**Step 3** Remove one RP and one FP from a box, using either OIR or CLI shutdown methods.

**Step 4** Configure application redundancy:

```bash
Router(config)# interface GigabitEthernet0/1/1
Router(config-if)# redundancy rii 600
Router(config-if)# redundancy group 1 ip 10.2.3.4 exclusive decrement 200
Router(config-if)# exit
Router(config)# redundancy
Router(config-red)# application redundancy
Router(config-red-app)# group 1
Router(config-red-app-grp)# name SBC
Router(config-red-app-grp)# data GigabitEthernet 0/0/1
Router(config-red-app-grp)# control GigabitEthernet 0/0/2 protocol 1
Router(config-red-app-grp)# timers delay 100 reload 400
Router(config-red-app-grp)# track 1 decrement 1
Router(config-red-app-grp)# track 2 decrement 1
Router(config-red-app-grp)# exit
Router(config-red-app)# protocol 1
Router(config-red-app-prtcl)# name BFD
Router(config-red-app-prtcl)# timers hello time 4 holdtime 6
Router(config-red-app-prtcl)# authentication md5 key-string 0 n1 100
```

**Step 5** Add the SBC application redundancy configuration after the RG is shutdown:

```bash
Router(config)# redundancy
Router(config-red)# application redundancy
Router(config-red-app)# group 1
Router(config-red-app-grp)# name SBC
Router(config-red-app-grp)# shutdown
Router(config-red-app-grp)# exit
Router(config-red-app)# exit
Router(config)# exit
Router(config)# sbc redundancy-group 1 tcp
Router(config)# redundancy
Router(config-red)# application redundancy
Router(config-red-app)# group 1
```
Chapter 15 Interchassis High Availability

Information About Interchassis High Availability

Step 6  
Save the SBC configuration and use the `no sbc` command to remove the SBC configuration:

```
Router(config)# no sbc ASR1
```

Step 7  
Check whether the Cisco ASR 1000 Series Router is in the ACTIVE mode or UNKNOWN mode because another Cisco ASR 1000 Series Router is not yet configured:

```
Router# show redundancy application transport group
```

Step 8  
Configure the SBC again using the saved configuration.

```
Router(config)# sbc ASR1
```

Step 9  
Place a test call to ensure that the SBC is functioning well.

Step 10  
Bring the second Cisco ASR 1000 Series Router online with a single RP and single FP.

Step 11  
Configure application redundancy on the second Cisco ASR 1000 Series Router:

```
Router(config)# interface GigabitEthernet0/1/1
Router(config-if)# redundancy group 1 ip 10.1.1.1 exclusive decrement 50
Router(config-if)# redundancy rii 10
Router(config-if)# exit
Router(config)# redundancy
Router(config-red)# application redundancy
Router(config-red-app)# group 1
Router(config-red-app-grp)# name SBC
Router(config-red-app-grp)# data GigabitEthernet 1/0/1
Router(config-red-app-grp)# control GigabitEthernet 0/0/1 protocol 1
Router(config-red-app-grp)# timers delay 100 reload 400
Router(config-red-app-grp)# track 1 decrement 1
Router(config-red-app-grp)# track 2 decrement 1
Router(config-red-app-grp)# exit
Router(config-red-app)# protocol 1
Router(config-red-app-prtcl)# name BFD
Router(config-red-app-prtcl)# timers hello 4 hold 6
Router(config-red-app-prtcl)# authentication md5 key-string 0 n1 100
```

Step 12  
Add the SBC application redundancy configuration to the second Cisco ASR 1000 Series Router after the RG is shut down:

```
Router(config)# redundancy
Router(config-red)# application redundancy
Router(config-red-app)# group 1
Router(config-red-app-grp)# name SBC
Router(config-red-app-grp)# shutdown
Router(config-red-app-grp)# exit
Router(config-red)# exit
Router(config)# sbc redundancy-group 1 tcp
```

Step 13  
Configure the second Cisco ASR 1000 Series Router such that it is in the STANDBY HOT mode:

```
Router# show redundancy application transport group
```
Step 14  Check whether the first Cisco ASR 1000 Series Router is still in the ACTIVE mode:

    Router# show redundancy application transport group

Step 15  Check whether the SBC configuration is synchronized to the Cisco ASR 1000 Series Router that is in the STANDBY mode:

    Router# show run

Step 16  Place a test call to check whether the SBC is still functioning.

### Interchassis to Intrachassis Conversion

The following steps describe the procedure involved in single RP box-to-box to dual RPs conversion:

---

**Step 1**  Configure two Cisco ASR 1000 Series Routers with single RP’s and single FP’s in the box-to-box mode.

**Step 2**  Generate test calls with multiple failovers to ensure proper box-to-box operation.

**Step 3**  Shut the RG on both the Cisco ASR 1000 Series Routers:

    Router(config)# redundancy
    Router(config-red)# application redundancy
    Router(config-red-app)# group 1
    Router(config-red-app-grp)# shutdown

**Step 4**  Remove the SBC redundancy configuration from both the Cisco ASR 1000 Series Routers:

    Router(config)# no sbc redundancy-group 1 tcp

**Step 5**  Remove the RG configuration from both the Cisco ASR 1000 Series Routers:

    Router(config-red)# no application redundancy

**Step 6**  Save the SBC configuration, and use the **no sbc** command to remove the SBC configuration:

    Router(config)# no sbc ASR1

**Step 7**  Add one RP and one FP to the Cisco ASR 1000 Series Router that is in the ACTIVE mode.

**Step 8**  Configure the SBC again using the saved configuration:

    Router(config)# sbc ASR1

**Step 9**  Check whether the SBC application of the primary Cisco ASR 1000 Series Router has been activated and is functioning correctly:

    Router# show redundancy application transport group

**Step 10**  Generate test calls to verify whether the SBC is functioning, and leave this call active in order to be able to perform the subsequent steps.

**Step 11**  Configure the SSO redundancy:

    Router(config)# redundancy
    Router(config-red)# mode sso

**Step 12**  Check whether the configuration is synchronized and there are no interruptions in the SBC traffic:

    Router# show run

**Step 13**  Place a test call to ensure that the SBC is still functioning.
Configuring Interchassis High Availability

To configure Interchassis High Availability, see the following sections in the “Configuring Firewall Stateful Inter-Chassis Redundancy” chapter of Security Configuration Guide: Zone-Based Policy Firewall Cisco IOS XE Release 3S:


This chapter provides information about the following topics:

• Configuring the Redundancy Application Group
• Configuring the Redundancy Group Protocol
• Configuring Virtual IP Address and Redundant Interface Identifier
• Configuring Control and Data Interface

Configuring Static Routing with Interchassis High Availability

When a static route is used in an upstream and downstream router or Layer 3 switch, VIP must be configured on the LAN-facing interface on the Cisco ASR 1000 Series Router. The static route that has an SBC interface IP address as the destination IP address sets the VIP address as the next hop address. Although this scenario offers the best convergence time during a failover, it faces an unicast flooding problem in the LAN between the router or the Layer 3 switch and the Cisco ASR 1000 Series Router. The default ARP table aging time is 4 hours, while the MAC table aging time is only a couple of minutes. A MAC aging timer, which is greater or equal to the ARP timeout, is required to prevent unicast flooding for both upstream LAN and downstream LAN. After the ARP table is timed out, it sends an ARP request towards the VIP. The active Cisco ASR 1000 Series Router replies to the ARP request with a VMAC. The MAC table is refreshed and the unicast flooding problem is resolved.

To increase the MAC aging timer or decrease the ARP aging timer for the VLAN with the unicast flooding problem, use one of the following commands on the router or the Layer 3 switch:

• The `arp timeout` command on a VLAN interface
• The `mac-address-table aging-time vlan` command

Configuring Dynamic Routing with Interchassis High Availability

The SBC interface must be included as part of the Open Shortest Path First (OSPF) area so that the SBC is advertised when the box becomes active. The following example shows an OSPF configuration, illustrating the SBC box-to-box application with routing:

```plaintext
router ospf 200
router-id 4.4.4.10
priority 11
nsf
network 4.4.0.0 0.0.255.255 area 0

interface SBC1
ip address 10.2.0.1 255.255.255.0 secondary
ip address 10.2.0.10 255.255.255.0 secondary
ip address 10.2.0.100 255.255.255.0
ip ospf 200 area 0
```
Assigning a Redundancy Group to the SBC

This task shows how to assign a redundancy group to the SBC:

**Note**
Configuration on the SBC interface is similar on both active and standby routers. However, redundancy group traffic interfaces have different IP addresses and a shared redundancy IP address. While performing this procedure on the Cisco ASR 1001 Router, Cisco ASR 1002 Router, and Cisco ASR 1004 Router, set the redundancy mode to `NONE`.

**SUMMARY STEPS**

1. `enable`
2. `configure terminal`
3. `redundancy`
4. `application redundancy`
5. `group id`
6. `shutdown`
7. `exit`
8. `exit`
9. `exit`
10. `sbc redundancy-group group-number tcp`
11. `redundancy`
12. `application redundancy`
13. `group id`
14. `no shutdown`
15. `end`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 <code>enable</code></td>
<td>Enables privileged EXEC mode. Enter your password if prompted.</td>
</tr>
</tbody>
</table>

**Example:**
`Router> enable`

| Step 2 `configure terminal`  | Enables the global configuration mode.          |

**Example:**
`Router# configure terminal`
### Command or Action

<table>
<thead>
<tr>
<th>Step 3</th>
<th>redundancy</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config)# redundancy</td>
</tr>
</tbody>
</table>

Enters the redundancy configuration mode.

<table>
<thead>
<tr>
<th>Step 4</th>
<th>application redundancy</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-red)# application redundancy</td>
</tr>
</tbody>
</table>

Enters the redundancy application configuration mode.

<table>
<thead>
<tr>
<th>Step 5</th>
<th>group id</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-red-app)# group 1</td>
</tr>
</tbody>
</table>

Enters the redundancy application group configuration mode.

- *id*—Specifies the redundancy group ID that ranges from 1 to 2.

<table>
<thead>
<tr>
<th>Step 6</th>
<th>shutdown</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-red-app-grp)# shutdown</td>
</tr>
</tbody>
</table>

To assign a redundancy group to the SBC, the redundancy group must be shut down.

<table>
<thead>
<tr>
<th>Step 7</th>
<th>exit</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-red-app-grp)# exit</td>
</tr>
</tbody>
</table>

Exits from the redundancy application group configuration mode and enters the redundancy application configuration mode.

<table>
<thead>
<tr>
<th>Step 8</th>
<th>exit</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-red-app)# exit</td>
</tr>
</tbody>
</table>

Exits from the redundancy application configuration mode and enters the redundancy configuration mode.

<table>
<thead>
<tr>
<th>Step 9</th>
<th>exit</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-red)# exit</td>
</tr>
</tbody>
</table>

Exits from the redundancy configuration mode and enters the global configuration mode.

<table>
<thead>
<tr>
<th>Step 10</th>
<th>sbc redundancy-group group-number tcp</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config)# sbc redundancy-group 1 tcp</td>
</tr>
</tbody>
</table>

Assigns the redundancy group to the SBC in order to track the following:

- *group-number*—Specifies the redundancy group number.
- *tcp*—Specifies the Transmission Control Protocol (TCP), and the redundancy group protocol.

<table>
<thead>
<tr>
<th>Step 11</th>
<th>redundancy</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config)# redundancy</td>
</tr>
</tbody>
</table>

Enters the redundancy configuration mode.

<table>
<thead>
<tr>
<th>Step 12</th>
<th>application redundancy</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-red)# application redundancy</td>
</tr>
</tbody>
</table>

Enters the redundancy application configuration mode.
Managing and Monitoring Interchassis High Availability

You can manage and monitor the Interchassis High Availability feature as explained in the following sections:

- Managing and Monitoring the Redundancy Group infrastructure, page 15-14
- Managing and Monitoring an SBC Redundancy Group, page 15-15

Managing and Monitoring the Redundancy Group infrastructure

To manage and monitor the redundancy group infrastructure, use the following commands:

- `redundancy application reload group group-number {peer | self}`—Forces an active redundancy group to reload, and a standby redundancy group to become the active redundancy group, without affecting the status of the active redundancy group.

- `show redundancy application {group-id | all}`—Shows the summary information pertaining to the specified group or all the groups.

- `show redundancy application faults {group-id | all}`—Shows information about the faults pertaining to the specified group or all the groups.

- `show redundancy application interface interface`—Shows the interface information associated with the redundancy groups.

- `show redundancy application protocol group-id`—Shows the protocol information pertaining to the specified group or all the groups.

- `show redundancy application transport {group-id | clients}`—Shows transport information pertaining to the specified group or all the groups.

To enable debug logging of the specified type of information associated with redundancy groups, use the following commands:

- `debug redundancy application vp {event | error}`
- `debug redundancy application transport {db | trace | event | error | timer}`
Managing and Monitoring an SBC Redundancy Group

To manage and monitor an SBC redundancy group, use the following commands:

- **show sbc name rg transport**—Shows the transport information pertaining to an SBC redundancy group.
- **show sbc name rg statistics**—Shows the transport statistics pertaining to an SBC redundancy group.
- **clear sbc name rg**—Clears the SBC redundancy group box-to-box statistics.
- **monitor event-trace sbc ha**—Configures event tracing pertaining to the SBC in order to include significant redundancy group events for generating the history for bootup and transition logs to assist in debugging.

The following example shows a sample output of the **show sbc name rg transport** command:

```
Router# show sbc MySBC rg transport
SBC HA RG connection parameters for domain 2
---------------------------------------------
Application Type        1
Handler                 53
My IP address           1.0.0.7
My L4 Port              1060
L3 Protocol             1
L4 Protocol             1
Peer IP address         1.0.0.6
Peer L4 Port            1060
My MTU                  1464
My L4 Offset            28
```

The following example shows a sample output of the **show sbc name rg statistics** command:

```
Router# show sbc MySBC rg statistics
SBC HA B2B statistics
---------------------------------------------
Number of messages successfully queued           = 407
Number of messages successfully sent             = 407
Number of IPS messages sent                     = 370
Number of messages queue failures               = 0
Number of attempted-send message failures        = 0
Number of message header malloc failures         = 0
Number of no packet available failures           = 0
Number of high watermark of queued messages      = 16
Number of high watermark of recv messages        = 15
Number of messages received                      = 412
Number of received IPS messages                  = 356
Number of received messages discarded            = 0
Number of received messages dropped(no group)    = 0
Number of received large IPS messages            = 37
Number of large message send failures            = 0
Number of large message send total               = 0
Number of large message recv failures            = 0
Number of large message not sent, unsupp by peer = 0
```
Upgrading Interchassis Redundancy

To upgrade interchassis redundancy, perform the following steps:

---

**Note**

Two Cisco ASR1000 Series Aggregation Services Routers are required to perform this procedure. While the primary router is the active one, the secondary router is the standby one.

---

**Step 1**

In the primary router, use the `show redundancy application group RG Group ID` command to display which router is the active one.

**Step 2**

In the secondary router, use the `show redundancy application group RG Group ID` command to display which router is the standby one.

**Step 3**

Download the latest version of the Cisco ASR 1000 Series Aggregation Services Routers image to both the primary router and the secondary router.

**Step 4**

On the secondary router, change the boot variable to the new image by using the `boot system bootflash: new image` command.

**Step 5**

On the primary router, synchronize the SBC by using the `sbc sbc name` command and the `sync` command. Wait for five minutes to make sure that the SBC configuration is fully synchronized to the standby router.

**Step 6**

On the secondary router, save the running configuration by using the `write memory` command.

**Step 7**

On the primary router, shut down the redundancy group.

The secondary router immediately becomes the active one, and all the active calls are preserved. Note that the router is still in service when switching over to the active router.

**Step 8**

On the primary router, change the boot variable to the new software image, and save the running configuration.

**Step 9**

Reload the primary router for upgrading, and wait for this router to come up with the upgraded version. It might take around 10 to 12 minutes from the time the router is reloaded.

**Step 10**

On the secondary router, shut down the redundancy and execute the `no shutdown` command for the redundancy group on the primary router as soon as possible.
Chapter 15      Interchassis High Availability

Step 11 The router will be down for around 100 seconds, and the primary router becomes active and in service with the upgraded software.

Step 12 Save the running configuration in the primary router.

Step 13 Reload the secondary router for upgrade. When you are asked whether you want to save the configuration before proceeding with the reload, enter No so that the secondary router will come up in the standby state after the upgrade.

The upgrade is completed.

Configuration Examples for Interchassis High Availability

To view the list of configuration examples pertaining to Interchassis High Availability, see the following sections in the “Configuring Firewall Stateful Inter-Chassis Redundancy” chapter of Security Configuration Guide: Zone-Based Policy Firewall Cisco IOS XE Release 3S at:


- Example: Configuring the Redundancy Application Group
- Example: Configuring the Redundancy Group Protocol
- Example: Configuring Virtual IP Address and Redundant Interface Identifier
- Example: Configuring Control and Data Interface
Fax Support

The Cisco Unified Border Element (SP Edition) media components enable fax over IP calls. Cisco Unified Border Element (SP Edition) supports the following types of fax over IP calls, using Session Initiation Protocol (SIP) or H.323:

- G.711 passthrough
- T.38 fax passthrough over the following protocols:
  - RTP: Real-time Transport Protocol
  - UDP-TL: A lightweight transport protocol for fax media that runs over User Datagram Protocol

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).

Feature History for Fax Support

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 2.4</td>
<td>This feature was introduced on the Cisco IOS XR, along with support for the unified model.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 2.5</td>
<td>The following were added in this release:</td>
</tr>
<tr>
<td></td>
<td>• Support for H.323</td>
</tr>
<tr>
<td></td>
<td>• G.711 passthrough support for SIP and H.323 interworking calls</td>
</tr>
<tr>
<td></td>
<td>• T.38 fax support for H.323 to H.323 and SIP to H.323 interworking calls</td>
</tr>
<tr>
<td></td>
<td>• Fax Upspeed support.</td>
</tr>
</tbody>
</table>

Contents

This module contains the following sections:

- Restrictions for Fax Support, page 16-2
- Fax Support, page 16-2
- Fax Upspeed Support, page 16-3
Restrictions for Fax Support

The following are restrictions for fax support in Cisco Unified Border Element (SP Edition):

- G.711 and T.38 interworking is not supported.
- T.38 fax passthrough does not support H.323 to SIP calls, or SIP to H.323 to SIP calls.
- Cisco proprietary fax is not supported, although it may work in the passthrough mode, because the Cisco Unified Border Element (SP Edition) does not police the RTP payload types, only the bandwidth. Cisco proprietary fax uses RTP.

Fax Support

Cisco Unified Border Element (SP Edition) supports two types of fax over IP:

- G.711 passthrough
- T.38 passthrough

G.711 Passthrough

G.711 passthrough is an International Telecommunication Union (ITU) standard codec operating at a 64 Kbps rate. It is a simple fax method and supports sending fax in the RTP stream of a typical G.711 call. G.711 is used by most VoIP providers because it provides high voice quality. It produces voice sounds similar to a regular or ISDN phone because G.711 does not use compression and uses the same codec used by the public switched telephone network (PSTN) and Integrated Services Digital Network (ISDN). Without the demand on processing power required by compression, G.711 has the lowest latency or lag.

Passthrough is a method of passing a FAX PCM stream across a VoIP network. It involves selecting a low-bandwidth codec (G.711), disabling silence suppression, and enabling echo cancellation. FAX passthrough is signalled by protocol stacks H.323 and SIP.

G.711 passthrough is supported for all cases of SIP and H.323 interworking calls in Cisco IOS XE Release 2.5 and later.

Note

The Cisco Unified Border Element (SP Edition)'s billing records for the call do not show anything explicitly because of the fax nature of the call. They merely report the standard set of metrics for the call, as they would do for a voice call.

T.38 Passthrough

T.38 Passthrough is an ITU standard for sending FAX across IP networks in a real-time mode. In Cisco Unified Border Element (SP Edition), T.38 fax calls are sent in-band using a fax-specific codec (rather than a general-purpose audio codec). T.38 fax uses a separately negotiated stream, which can either be negotiated at the start of the call (bandwidth will be reserved for it at that point), or renegotiated during the call (which may fail).

Passthrough is a method of passing a FAX PCM stream across a VoIP network. It involves selecting a high-bandwidth codec, disabling silence suppression, and enabling echo cancellation. FAX passthrough is signalled by protocol stacks H.323 and SIP.
Fax Support

T.38 fax passthrough is supported for SIP to SIP calls in Cisco IOS XE Release 2.4 and later. Support for SIP to H.323 and H.323 to H.323 calls is added in Cisco IOS XE Release 2.5 and later. In the case of SIP to H.323 calls, only the SIP side of the call will initiate the T.38 passthrough.

T.38 fax passthrough does not support H.323 to SIP calls or SIP to H.323 to SIP calls.

---

**Note**
The bandwidth reserved for a T.38 call is considered sufficient for carrying a T.38 rate of 14,400 bits per second and does not reflect the signaled rate in T.38.

---

**Note**
If an unnumbered datagram protocol transport layer (UDPTL) error correction is used for T.38, then the bandwidth reservation also includes capacity for up to three redundant parity packets in the T.38 stream.

### Fax Upspeed Support

Cisco Unified Border Element (SP Edition) supports fax upspeed. Fax upspeed is the ability of the SBC to change a codec in midcall by re-negotiation. The fax upspeed function is only supported when one of the endpoints engaged in the call initiates it; the SBC does not initiate the upspeed action. The SBC is capable of handling midcall codec re-negotiations to and from either H.323 or SIP interfaces.

When an endpoint has determined that the call is a fax or data call and calculates that the codec negotiated is too highly compressed to reliably pass tones, it may initiate a re-Invite to perform a codec re-negotiation that offers the G.711 codec, a higher bandwidth codec. Thus the process of re-negotiating to a higher bandwidth codec is called “fax upspeed.” The G.711 codec is also known as PCMA/PCMU.

When the endpoint determines that the fax or data call has ended, then the endpoint can send another Invite re-negotiation to switch back to a lower bandwidth codec.

The same mechanism applies to H.323 call legs, where there is a terminal capability exchange (TCS) and media channels are closed and new ones reopened.
Codec Handling

A compressor-decompressor (codec) is a device or program that performs a transformation on a data stream or signal. Cisco Unified Border Element (SP Edition) is hard-coded with a set of recognized codecs (see Table 17-1 to Table 17-5), including all commonly used voice and video codecs. The default behavior is to allow all recognized codecs on all calls. Any other codec present in call signaling is removed by Cisco Unified Border Element (SP Edition).

This enhancement allows you to restrict which codec(s) a particular call can use and to configure a minimum permissible packetization period for each permitted codec.

Cisco Unified Border Element (SP Edition) supports passthrough codecs that are passed through the SBC without transcoding. See Table 17-4 on page 17-7 for a list of passthrough codecs by type.

H.323 only supports PCMA, PCMU, G.722, G.723, G.728, G.729, GSM, telephone-event, H.261, H.263, H.264, and T.38 codecs. Therefore in a SIP to H.323 call, if SIP codecs are sent to H.323 that H.323 cannot support, then these codecs are not passed through and the call may fail accordingly.

Cisco Unified Border Element (SP Edition) supports codec transcoding for SIP to SIP calls using an external DSP resource or transcoding resource. Transcoding is the process of translating a media stream encoded using one codec into a media stream encoded using another codec, for example, translating a media stream encoded as PCMU into one encoded as G.726-32. For more information on transcoding, see the Implementing Transcoding chapter.

Cisco Unified Border Element (SP Edition) enables H.323 TCS Codecs support in Cisco IOS XE Release 2.5.1. This support provides the ability to announce media capabilities on behalf of a SIP endpoint to an H.323 endpoint by adding extra offered codecs in the H.245 Terminal Capability Set (TCS) message. See the H.323 TCS Codecs Support section on page 17-16.

Note

For Cisco IOS XE Release 2.4 and later, this feature is supported in the unified model.

For a complete description of the commands used in this chapter, refer to the Cisco Unified Border Element (SP Edition) Command Reference: Unified Model at:


For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.
Feature History for Codec Handling

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 2.4</td>
<td>This feature was introduced on the Cisco IOS XR along with support for the unified model.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 2.4.2</td>
<td>Support for packetization time, p-time, attribute in the SDP offer or answer configured in a codec list was added.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 2.5</td>
<td>Support for H.323 calls, and H.323 video codecs H.261, H.263, and H.264 were added. Support for passthrough codecs without transcoding was added.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 2.5.1</td>
<td>H.323 TCS Codecs support was added.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 2.6</td>
<td>Support for Dynamic CODEC configuration, multiple audio codec, and multiple video codec were added. H.323 support for clear channel data and modem calls was added.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 3.2S</td>
<td>Codec preference and reordering support was added.</td>
</tr>
</tbody>
</table>

Contents

This module contains the following sections:

- Prerequisites for Codec Handling, page 17-3
- Restrictions for Codec Handling, page 17-3
- Restriction for H.323 TCS Codecs Support, page 17-3
- Restrictions for Dynamic Codec Configuration, page 17-4
- Restrictions for Codec Reordering, page 17-4
- Codec Handling and Restriction, page 17-4
- Dynamic Codec Configuration, page 17-8
- Configuring Codec Restriction, page 17-9
- Packetization Time, page 17-15
- H.323 TCS Codecs Support, page 17-16
- Codecs Preference and Reordering Support, page 17-17
- Configuration Examples—Configuring Codec Restriction, page 17-22
- Configuration Examples—H.323 TCS Codecs Support, page 17-24
- Configuration Example—Defining a Codec using Dynamic Codec Configuration, page 17-25
Prerequisites for Codec Handling

The following prerequisite is required before you can restrict codecs:

- Before implementing Codec Handling, Cisco Unified Border Element (SP Edition) must already be configured.
- All signaling border element (SBE) and data border element (DBE) configurations required to make simple calls must already be configured.

Restrictions for Codec Handling

Review the following restrictions for codecs:

- For H.323 calls, SIP to H.323 and H.323 to SIP calls, both the callee and caller must use the same codec because any calls requiring transcoding will fail the call setup.
- The media packet forwarder on the DBE polices the bandwidth consumed by each media stream, but it cannot police the type of codecs or the packetization periods.
- Unrecognized codecs cannot be configured as members of the codec whitelist.
- Active calls are not released if there is a change in the codec whitelist during the call.
- If a codec whitelist is configured, Cisco Unified Border Element (SP Edition) removes any unlisted codecs from the call setup flow and media gate allocation.
- Multiple codec whitelists can be configured on a Call Admission Control (CAC) policy basis. For example, the list of codecs allowed for calls from SipAdj1 can be different than the list of codecs allowed for calls from SipAdj2.
- If a codec whitelist has not been configured, all recognized codecs (see Table 17-1 to Table 17-5) are allowed for all calls.
- You must use the textual value of the codec description that appears on the Session Description Protocol (SDP) to configure the codec whitelist, for example “PCMU” or “telephone-event”.
- Disallowing all codecs is not supported. However, you could set a bandwidth limit of 0 to achieve the same result.
- Codec lists are not applied to media-bypass calls (those in which Cisco Unified Border Element (SP Edition) does not reserve media resources).
- The format of the codec name is the same as the string used to represent it in SDP, for example PCMU or VDVI. All recognized codec names are listed in Table 17-1 – Table 17-5.
- A single codec can only be added to each list once, with a single packetization period.
- For each codec on a list, CAC restricts the signaled packetization period for any stream using that codec to be greater than or equal to the packetization period configured along with the codec in the list. If a stream uses more than one codec in the list, then the greater of all the packetization periods configured for each codec in the list is applied to the stream.

Restriction for H.323 TCS Codecs Support

H.323 TCS Codecs support only applies to H.323-to-SIP or SIP-to-H.323 interworking calls with the caller or callee using SIP.
Restrictions for Dynamic Codec Configuration

The following restrictions apply for Dynamic codec configuration feature:

- Codecs provided with the SBC (system codec) may be modified but not deleted.
- Codec names are case insensitive.
- A maximum of 100 user defined codecs is supported.
- Custom codec payload ID should always be 96.
- When defining new codec, codec type must be specified before all other fields.
- A custom codec is not defined until a codec type is specified.
- The ID of a system codec should not be changed.
- The ID should be unique for a Codec.
- A custom codec should not be deleted if used in a codec list.
- Sample size is only valid for sampling codecs.
- Bandwidth cannot be specified for redundancy or sampling codecs.
- Channels may only be specified for sampling based codecs.

Restrictions for Codec Reordering

- SBC supports codec reordering only for SIP-SDP codecs. Reordering of codecs in H.323 is not supported.
- If media bypass is configured, codec reordering cannot be applied.

Codec Handling and Restriction

Note

The bandwidths listed in the tables below are the bandwidths without the transport layer overheads. Therefore, the actual bandwidths reserved by Cisco Unified Border Element (SP Edition) are higher than the listed values.

Table 17-1 lists sample-based audio codecs.

Table 17-1 Sample-Based Audio Codecs with Packetization Time 10 ms

<table>
<thead>
<tr>
<th>Payload Type</th>
<th>Codec Name</th>
<th>Clock Rate (Hz)</th>
<th>Sample Size (bits)</th>
<th>Channels</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>PCMU (also known as G.711)</td>
<td>8000</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>DVI4</td>
<td>8000</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>DVI4</td>
<td>16000</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>PCMA</td>
<td>8000</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>L16</td>
<td>44100</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td>L16</td>
<td>44100</td>
<td>16</td>
<td>1</td>
</tr>
</tbody>
</table>
Table 17-1  Sample-Based Audio Codecs with Packetization Time 10 ms (continued)

<table>
<thead>
<tr>
<th>Payload Type</th>
<th>Codec Name</th>
<th>Clock Rate (Hz)</th>
<th>Sample Size (bits)</th>
<th>Channels</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>G728</td>
<td>8000</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>16</td>
<td>DVI4</td>
<td>11025</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>17</td>
<td>DVI4</td>
<td>22050</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>—</td>
<td>G726-40</td>
<td>8000</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>G726-32</td>
<td>8000</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>—</td>
<td>G726-24</td>
<td>8000</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>—</td>
<td>G726-16</td>
<td>8000</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>—</td>
<td>L8</td>
<td>8000</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>—</td>
<td>DAT12</td>
<td>8000</td>
<td>12</td>
<td>2</td>
</tr>
<tr>
<td>—</td>
<td>L20</td>
<td>44100</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>—</td>
<td>L24</td>
<td>44100</td>
<td>24</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 17-2 lists non-sample-based audio codecs.

Table 17-2  Non-Sample-Based Audio Codecs

<table>
<thead>
<tr>
<th>Payload Type</th>
<th>Codec Name</th>
<th>Packetization Time (ms)</th>
<th>Allocated Bandwidth (bps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>GSM</td>
<td>10</td>
<td>13200</td>
</tr>
<tr>
<td>4</td>
<td>G723</td>
<td>30</td>
<td>6400</td>
</tr>
<tr>
<td>7</td>
<td>LPC</td>
<td>10</td>
<td>5600</td>
</tr>
<tr>
<td>9</td>
<td>G722</td>
<td>10</td>
<td>64000</td>
</tr>
<tr>
<td>12</td>
<td>QCELP</td>
<td>—</td>
<td>13300</td>
</tr>
<tr>
<td>13</td>
<td>CN</td>
<td>10</td>
<td>400</td>
</tr>
<tr>
<td>14</td>
<td>MPA</td>
<td>N/A</td>
<td>131072</td>
</tr>
<tr>
<td>18</td>
<td>G729</td>
<td>10</td>
<td>8000</td>
</tr>
<tr>
<td>18</td>
<td>G.729A</td>
<td>10</td>
<td>8000</td>
</tr>
<tr>
<td>—</td>
<td>G729B</td>
<td>20</td>
<td>8000</td>
</tr>
<tr>
<td>—</td>
<td>G729AB</td>
<td>10</td>
<td>8000</td>
</tr>
<tr>
<td>—</td>
<td>G729D</td>
<td>10</td>
<td>6400</td>
</tr>
<tr>
<td>—</td>
<td>G729E</td>
<td>10</td>
<td>11800</td>
</tr>
<tr>
<td>—</td>
<td>GSM-EFR</td>
<td>10</td>
<td>12400</td>
</tr>
<tr>
<td>—</td>
<td>iSAC</td>
<td>30</td>
<td>32000</td>
</tr>
<tr>
<td>—</td>
<td>VDVI</td>
<td>10</td>
<td>25000</td>
</tr>
<tr>
<td>—</td>
<td>AMR</td>
<td>10</td>
<td>12500</td>
</tr>
<tr>
<td>—</td>
<td>AMR-WB</td>
<td>10</td>
<td>24420</td>
</tr>
<tr>
<td>—</td>
<td>dsr-es201108</td>
<td>10</td>
<td>4800</td>
</tr>
<tr>
<td>—</td>
<td>EVRC</td>
<td>10</td>
<td>8550</td>
</tr>
</tbody>
</table>
Codec Handling and Restriction

Chapter 17 Codec Handling

Table 17-2 Non-Sample-Based Audio Codecs (continued)

<table>
<thead>
<tr>
<th>Payload Type</th>
<th>Codec Name</th>
<th>Packetization Time (ms)</th>
<th>Allocated Bandwidth (bps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>EVRC0</td>
<td>10</td>
<td>8550</td>
</tr>
<tr>
<td>—</td>
<td>mpa-robust</td>
<td>10</td>
<td>327680</td>
</tr>
<tr>
<td>—</td>
<td>G7221</td>
<td>10</td>
<td>32000</td>
</tr>
<tr>
<td>—</td>
<td>MP4A-LATM</td>
<td>10</td>
<td>131072</td>
</tr>
<tr>
<td>—</td>
<td>SMV</td>
<td>10</td>
<td>8550</td>
</tr>
<tr>
<td>—</td>
<td>SMV0</td>
<td>10</td>
<td>8550</td>
</tr>
</tbody>
</table>

Table 17-3 lists video codecs.

Table 17-3 Video Codecs

<table>
<thead>
<tr>
<th>Payload Type</th>
<th>Codec Name</th>
<th>Packetization Time (ms)</th>
<th>Allocated Bandwidth (bps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>CelB</td>
<td>—</td>
<td>524228</td>
</tr>
<tr>
<td>26</td>
<td>JPEG</td>
<td>—</td>
<td>524228</td>
</tr>
<tr>
<td>28</td>
<td>nv</td>
<td>—</td>
<td>524228</td>
</tr>
<tr>
<td>31</td>
<td>H261</td>
<td>—</td>
<td>524228</td>
</tr>
<tr>
<td>32</td>
<td>MPV</td>
<td>—</td>
<td>524228</td>
</tr>
<tr>
<td>33</td>
<td>MP2T</td>
<td>—</td>
<td>524228</td>
</tr>
<tr>
<td>34</td>
<td>H263</td>
<td>—</td>
<td>524228</td>
</tr>
<tr>
<td>NA</td>
<td>H264</td>
<td>—</td>
<td>210000000</td>
</tr>
<tr>
<td>—</td>
<td>BMPEG</td>
<td>—</td>
<td>524228</td>
</tr>
<tr>
<td>—</td>
<td>BT656</td>
<td>—</td>
<td>170000000</td>
</tr>
<tr>
<td>—</td>
<td>DV</td>
<td>—</td>
<td>1500000000</td>
</tr>
<tr>
<td>—</td>
<td>H263-1998</td>
<td>—</td>
<td>524228</td>
</tr>
<tr>
<td>—</td>
<td>H263-2000</td>
<td>—</td>
<td>524228</td>
</tr>
<tr>
<td>—</td>
<td>MP1S</td>
<td>—</td>
<td>1600000</td>
</tr>
<tr>
<td>—</td>
<td>MP2P</td>
<td>—</td>
<td>524228</td>
</tr>
<tr>
<td>—</td>
<td>MP4V-ES</td>
<td>—</td>
<td>524228</td>
</tr>
<tr>
<td>—</td>
<td>raw</td>
<td>N/A</td>
<td>1500000000</td>
</tr>
<tr>
<td>—</td>
<td>SMPTE292M</td>
<td>N/A</td>
<td>1500000000</td>
</tr>
</tbody>
</table>

Table 17-4 lists the supported passthrough codecs without transcoding.
## Codec Handling and Restriction

### Table 17-4  Passthrough Codecs without Transcoding

<table>
<thead>
<tr>
<th>Codec Name</th>
<th>Codec Type</th>
<th>Packetization Time (ms)</th>
<th>Allocated Bandwidth (bps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCMA (also known as G.711)</td>
<td>sample-based audio</td>
<td>10</td>
<td>64000</td>
</tr>
<tr>
<td>PCMU (also known as G.711)</td>
<td>sample-based audio</td>
<td>10</td>
<td>64000</td>
</tr>
<tr>
<td>G726-16</td>
<td>audio</td>
<td>10</td>
<td>NA</td>
</tr>
<tr>
<td>G726-24</td>
<td>audio</td>
<td>10</td>
<td>NA</td>
</tr>
<tr>
<td>G726-32</td>
<td>audio</td>
<td>10</td>
<td>NA</td>
</tr>
<tr>
<td>G728</td>
<td>audio</td>
<td>10</td>
<td>NA</td>
</tr>
<tr>
<td>G729 A</td>
<td>non-sample-based audio</td>
<td>10</td>
<td>8000</td>
</tr>
<tr>
<td>G729 B</td>
<td>non-sample-based audio</td>
<td>10</td>
<td>8000</td>
</tr>
<tr>
<td>G723-53</td>
<td>non-sample-based audio</td>
<td>30</td>
<td>6400</td>
</tr>
<tr>
<td>G723-63</td>
<td>non-sample-based audio</td>
<td>30</td>
<td>6400</td>
</tr>
<tr>
<td>GSM/GSM-FR</td>
<td>non-sample-based audio</td>
<td>10</td>
<td>13200</td>
</tr>
<tr>
<td>GSM-EFR</td>
<td>non-sample-based audio</td>
<td>20</td>
<td>12400</td>
</tr>
<tr>
<td>GSM-HR</td>
<td>non-sample-based audio</td>
<td>20</td>
<td>5600</td>
</tr>
<tr>
<td>AMR</td>
<td>non-sample-based audio</td>
<td>10</td>
<td>12500</td>
</tr>
<tr>
<td>EVRC</td>
<td>non-sample-based audio</td>
<td>10</td>
<td>8550</td>
</tr>
<tr>
<td>G722</td>
<td>non-sample-based audio</td>
<td>10</td>
<td>64000</td>
</tr>
<tr>
<td>iLBC</td>
<td>non-sample-based audio</td>
<td>20</td>
<td>15200</td>
</tr>
<tr>
<td>H.261</td>
<td>video</td>
<td>NA</td>
<td>524228</td>
</tr>
<tr>
<td>H.263</td>
<td>video</td>
<td>NA</td>
<td>524228</td>
</tr>
</tbody>
</table>

Table 17-5 lists other audio codecs.

### Table 17-5  Other Codecs

<table>
<thead>
<tr>
<th>Codec Name</th>
<th>Packetization Time (ms)</th>
<th>Allocated Bandwidth (bps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>telephone-event</td>
<td>20</td>
<td>1600</td>
</tr>
<tr>
<td>tone</td>
<td>20</td>
<td>1600</td>
</tr>
<tr>
<td>RED</td>
<td>20</td>
<td>1</td>
</tr>
<tr>
<td>parityfec</td>
<td>20</td>
<td>1</td>
</tr>
<tr>
<td>T140</td>
<td>100</td>
<td>80</td>
</tr>
<tr>
<td>pointer</td>
<td>20</td>
<td>1600</td>
</tr>
<tr>
<td>H224</td>
<td>20</td>
<td>6560</td>
</tr>
<tr>
<td>T38</td>
<td>N/A</td>
<td>15500</td>
</tr>
<tr>
<td>X-NSE</td>
<td>20</td>
<td>1600</td>
</tr>
</tbody>
</table>
Dynamic Codec Configuration

The Dynamic Codec configuration feature allows you to:

- **Define new codecs**—You can create variants of codecs that are included in the SBC. For example, G.729 A is a codec that is included in SBC. You may define a variant called G.729.1 using Dynamic Codec feature.

- **Modify existing codecs**—You can modify certain attributes of the codecs included in the SBC. For example, you can change the bandwidth characteristics of H.264 video codecs, included in the SBC.

- **Display codecs supported on SBC**—You can view all codecs supported on an SBC.

---

**Note**
Aliasing and Codec conversion are not supported by Dynamic Codec Configuration feature in this release.

Audio and Video codecs that are not included in SBC can be defined using the Dynamic Codec Configuration feature.

---

Multiple Audio Codec Support

SBC transparently passes the following codecs:

- G.711 with silence suppression and RFC 2833 (and RFC 4733)
- G.723
- G.726
- G.726 with silence suppression (Silence suppression is used to save bandwidth by not sending the data when there is no voice during a call.)
- G.729
- G.729a
- G.729a/b
- AMR
- AAC-LD
- AMR-WB
- AMR-WB+
- G.718

Audio codecs not included in SBC can be defined using the Dynamic Codec Configuration feature.

---

Multiple Video Codec Support

SBC transparently passes media encoded using:

- H.263
- H.264
- H.264 AVC
Video codecs not included in SBC can be defined using the Dynamic Codec Configuration feature.

**H.323 Support for Clear Channel Data/Modem Calls**

The clearmode codec is supported in existing VoIP equipments, and is used in the H.323, SIP, and MGCP signaling protocols. The Cisco SBC supports the clearmode codec in all places where it supports common audio codecs, such as G.711.

**Configuring Codec Restriction**

You first configure the codecs and then apply them as explained in the following sections:

- Configuring Codecs, page 17-9
- Configuring a CAC Policy to Use a Codec List, page 17-11

**Configuring Codecs**

To restrict which codec(s) a particular call can use and to configure a minimum permissible packetization period for each permitted codec, you must configure CAC with a list of codecs, provide a description for the list, and then add any codec(s) to the list.

**SUMMARY STEPS**

1. configure terminal
2. sbc service-name
3. sbe
4. codec list name
5. description text
6. codec codec-name [packetization-period packetization-period]
7. end
8. show sbc service-name sbe codec-list list-name
## Configuring Codec Restriction

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc service-name</td>
<td>Enters the submode of an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# sbc mysbc</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Configures the submode of the SBE entity within a SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> codec list name</td>
<td>Creates a codec list.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe)# sbc mysbc sbe codec list my_codecs</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> description text</td>
<td>Adds a description for the specified codec list using a readable text string format.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-codec-list)# description Legitimate codes</td>
<td>The no form of this command removes the description. This description is displayed when the show command is used for this codec list. It is also included for each codec list when a summary of all codec lists is displayed.</td>
</tr>
<tr>
<td><strong>Step 6</strong> codec codec-name [packetization-period packetization-period]</td>
<td>Adds a codec to a codec list, and sets a minimum packetization period (optional) for the codec.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-codec-list)# codec PCMU packetization-period 20</td>
<td>The no form of this command (without the packetization period) removes the named codec from the codec list. Note If the no form of this command includes the packetization period, only the packetization period for the codec is removed.</td>
</tr>
<tr>
<td><strong>Step 7</strong> end</td>
<td>Exits codec-list mode and enters Privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-codec-list)# end</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> show sbc service-name sbe codec-list list-name</td>
<td>Displays detailed information about the codec lists configured on the SBE.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# show sbc mysbc sbe codec-list my_codecs</td>
<td>If the list name is omitted, for example, my_codecs, then details are displayed for all codec lists on the SBE.</td>
</tr>
</tbody>
</table>
Configuring a CAC Policy to Use a Codec List

SUMMARY STEPS

1. configure terminal
2. sbc service-name
3. sbe
4. cac-policy-set policy-set-id
5. first-cac-scope scope-name
6. first-cac-table table-name
7. cac-table table-name
8. table-type {policy-set | limit {list of limit tables}}
9. entry entry-id
10. cac-scope {list of scope options}
11. action [next-table goto-table-name | cac-complete]
12. codec-restrict-to-list list-name
13. complete
14. end
15. show sbc service-name sbe cac-policy-set id table name entry entry

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc service-name</td>
<td>Enters the submode of an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# sbc mySBC</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Configures the submode of the SBE entity within a SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> cac-policy-set policy-set-id</td>
<td>Enters the submode of CAC policy.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe)# cac-policy-set 1</td>
<td></td>
</tr>
</tbody>
</table>
### Configuring Codec Restriction

**Step 5**

**Command or Action:**

```
first-cac-scope scope-name
```

**Example:**

```
Router(config-sbc-sbe-cacpolicy)# first-cac-scope global
```

**Purpose:**

Configures the scope at which to begin defining limits when performing the admission control stage of policy.

**Note:** The first-cac-scope definition is only relevant if the table type configured by the first-cac-table command is a Limit table. In that case, the scope of the first-cac-table is determined by first-cac-scope. If the first-cac-table is a Policy Set table, the first-cac-scope is ignored and defaults to global.

The `scope-name` argument configures the scope at which limits should be initially defined. Possible values are:

- adj-group
- call
- category
- dst-account
- dst-adj-group
- dst-adjacency
- dst-number
- global
- src-account
- src-adj-group
- src-adjacency
- src-number

Features can be enabled or disabled per adjacency group through CAC configuration the same way this is done per individual adjacencies.

**Step 6**

**Command or Action:**

```
first-cac-table table-name
```

**Example:**

```
Router(config-sbc-sbe-cacpolicy)# first-cac-table first_policy_table
```

**Purpose:**

Configures the name of the first policy table to process when performing the admission control stage of policy.

**Step 7**

**Command or Action:**

```
cac-table table-name
```

**Example:**

```
Router(config-sbc-sbe-cacpolicy)# cac-table first_policy_table
```

**Purpose:**

Enters the mode for configuration of an admission control table (creating one if necessary) within the context of an SBE policy set.
### Command or Action

| Step 8 | table-type {policy-set | limit (list of limit tables)} |
|--------|--------------------------------------------------|

**Example:**

```bash
Router(config-sbc-sbe-cacpolicy-cactable)#
table-type policy-set
```

### Purpose

Configures the table type of a CAC table within the context of an SBE policy set.

The `list of limit tables` argument controls the syntax of the match-value fields of the entries in the table. Possible available Limit tables are:

- **account**—Compare the name of the account.
- **adj-group**—Compare the name of the adjacency group.
- **adjacency**—Compare the name of the adjacency.
- **all**—No comparison type. All events match this type.
- **call-priority**—Compare with call priority.
- **category**—Compare the number analysis assigned category.
- **dst-account**—Compare the name of the destination account.
- **dst-adj-group**—Compare the name of the destination adjacency group.
- **dst-adjacency**—Compare the name of the destination adjacency.
- **dst-prefix**—Compare the beginning of the dialed digit string.
- **event-type**—Compare with CAC policy event types.
- **src-account**—Compare the name of the source account.
- **src-adj-group**—Compare the name of the source adjacency group.
- **src-adjacency**—Compare the name of the source adjacency.
- **src-prefix**—Compare the beginning of the calling number string.

**Note**  
For Limit tables, the event or message or call matches only a single entry.

Features can be enabled or disabled per adjacency group through CAC configuration the same way this is done per individual adjacencies. The adj-group table type matches on either source or destination adjacency group.

When the policy-set keyword is specified, use the `cac-scope` command to configure the scope within each entry at which limits are applied in a CAC Policy Set table.

**Note**  
For Policy Set tables, the event or call or message is applied to all entries in this table.
## Command or Action

<table>
<thead>
<tr>
<th>Step 9</th>
<th>entry entry-id</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-cacpolicy-cactable)# entry 1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 10</th>
<th>cac-scope (list of scope options)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-cacpolicy-cactable-entry)# cac-scope call</td>
<td></td>
</tr>
</tbody>
</table>

## Purpose

Enters the mode to create or modify an entry in an admission control table.

Configures the scope within each of the entries at which limits are applied in a policy set table.

- *list of scope options*—Specifies one of the following strings used to match events:
  - account—Events that are from the same account.
  - adjacency—Events that are from the same adjacency.
  - adj-group—Events that are from members of the same adjacency group.
  - call—Scope limits are per single call.
  - category—Events that have same category.
  - dst-account—Events that are sent to the same account.
  - dst-adj-group—Events that are sent to the same adjacency group.
  - dst-adjacency—Events that are sent to the same adjacency.
  - dst-number—Events that have same destination.
  - global—Scope limits are global
  - src-account—Events that are from the same account.
  - src-adj-group—Events that are from the same adjacency group.
  - src-adjacency—Events that are from the same adjacency.
  - src-number—Events that have the same source number.
  - sub-category—The limits specified in this scope apply to all events sent to or received from members of the same subscriber category.
  - sub-category-pfx—The limits specified in this scope apply to all events sent to or received from members of the same subscriber category prefix.
  - subscriber—The limits specified in this scope apply to all events sent to or received from individual subscribers (a device that is registered with a Registrar server).
Packetization Time

The packetization time (p-time) is the time period a codec is applied to a media stream to build a single digital packet. When a call travels between two endpoints, both the codec and the p-time for that codec is negotiated. The SBC acts on the p-time attribute in the SDP based on the configuration in the codec list.

SBC adds the ptime attribute to an SDP offer or answer, when one is configured explicitly in a codec whitelist. The exception is if a minimum packetization time is configured as part of SBC policy. This configuration causes SBC to insert a=ptime lines into forwarded offers or answers.

SBC ensures that media clipping does not occur as a result of overestimating the ptime (and hence underestimating the bandwidth requirement).

---

### Command or Action

**Step 11**

`action [next-table goto-table-name | cac-complete]`

**Example:**

```bash
Router(config-sbc-sbe-cacpolicy-cactable-entry)
# action cac-complete
```

**Purpose:**

Configures the action to perform after this entry in an admission control table. Possible actions are:

- Identify the next CAC table to process using the `next-table` keyword and the `goto-table-name` argument.
- Stop processing for this scope using the `cac-complete` keyword.

**Step 12**

`codec-restrict-to-list list-name`

**Example:**

```bash
Router(config-sbc-sbe-cacpolicy-cactable-entry)
# codec-restrict-to-list my_codecs
```

**Purpose:**

Configures CAC to restrict the codecs used in signaling a call to the set of codecs provided in the named list.

- If a codec list is empty, all codecs recognized by the SBE are allowed.
- The `no` form of this command, or not setting this command, allows any recognized codecs to be used without restrictions.

**Note**

This command replaces any codec list that was set up by an earlier CAC entry. To clear all restrictions from an earlier CAC entry, you must configure a `codec-restrict-to-list list-name`, where `list-name` is the name of a list containing no codecs.

**Step 13**

`complete`

**Example:**

```bash
Router(config-sbc-sbe-cacpolicy)# complete
```

**Purpose:**

Completes the cac-policy.

**Step 14**

`end`

**Example:**

```bash
Router(config-sbc-sbe-cacpolicy)# end
```

**Purpose:**

Exits the cac-policy-set configuration mode and enters Privileged EXEC mode.

**Step 15**

`show sbc service-name sbe cac-policy-set id table name entry entry`

**Example:**

```bash
Router# show sbc mysbc sbe cac-policy-set 1 table standard_policy_list entry 1
```

**Purpose:**

Displays detailed information for a specific entry in a CAC policy table, including any restricted codecs.
If ptime is signaled explicitly on the offer and answer, it uses the lower of the two values to calculate the bandwidth allowance for both directions of media.

## H.323 TCS Codecs Support

Cisco Unified Border Element (SP Edition) enables H.323 TCS Codecs support in Cisco IOS XE Release 2.5.1. This support provides the ability to announce media capabilities on behalf of a SIP endpoint to an H.323 endpoint by adding extra offered codecs in the H.245 Terminal Capability Set (TCS) message. You configure the additional codecs in the TCS message with Call Admission Control (CAC) policy commands.

H.323 TCS Codecs support applies to SIP to H.323 or H.323 to SIP interworking calls when the caller or callee uses SIP.

### Information About H.323 TCS Codecs Support

H.323 TCS Codecs support adds extra capability in the H.245 Terminal Capability Set (TCS) message by providing the ability to configure offered codecs in the TCS message, on behalf of a SIP endpoint for SIP to H.323 or H.323 to SIP interworking calls when the caller or callee uses SIP.

A TCS message is an H.245 message transmitted by an H.323 endpoint sent by the SBC. The TCS message transmitted by the H.323 endpoint is used to indicate the H.323 endpoint’s media capabilities, as well as the version of H.245, to the other party.

However, a SIP endpoint's SDP offer and answer may not announce all its capabilities because the SDP offer may be missing T.38 and other codecs, and the SDP answer may be missing many codecs. For example, a call originated by SIP endpoint A may not indicate the support of T.38, which would cause the H.323 endpoint B to be unaware of the capability. Therefore, endpoint B would not be able to send a RequestMode message to the SBC asking endpoint A to switch to the T38 codec.

The H.323 TCS Codecs support feature overcomes this gap by allowing you to add offered codecs in the TCS message that will announce the SIP caller or callee media capabilities to the H.323 endpoint. You do this with the `caller-media-caps` or `callee-media-caps` command in CAC table entry configuration mode. These two commands configure the SIP caller or callee media capability by assigning a codec list that is used to announce media capabilities on behalf of either a SIP caller or callee in a SIP to H.323 or H.323 to SIP interworking call.

You can also use the similar `tcs-extra-codecs` command in a CAC table entry to configure additional codecs in a codec list that sends an extra TCS message to the H.323 side. This command announces extra codecs capability to the H.323 endpoint on behalf of the SIP side, whether it is the SIP caller or SIP callee in a SIP to H.323 or H.323 to SIP interworking call.

You can use any one of the three commands independently.

---

**Note**

In some scenarios, the `branch` command can be used as an alternative to the `caller` and `callee` commands. The `branch` command has been introduced in Release 3.5.0. See the "Configuring Directed Nonlimiting CAC Policies?" section on page 7-37 for information about this command.

---

You can also use the similar `tcs-extra-codecs` command in a CAC table entry to configure additional codecs in a codec list that sends an extra TCS message to the H.323 side. This command announces extra codecs capability to the H.323 endpoint on behalf of the SIP side, whether it is the SIP caller or SIP callee in a SIP to H.323 or H.323 to SIP interworking call.

You can use any one of the three commands independently.

---

**Note**

After a codec list has been assigned, it may not be deleted until it is removed from the CAC entry. A codec list must exist before it can be assigned to an entry in a CAC table.
Codecs Preference and Reordering Support

SBC supports reordering of the codecs in a list, and assigning a priority to each codec. You can also apply a codec preference to a list in the CAC policy entry.

For information about restrictions for the Codecs Preference and reordering feature, see the Restrictions for Codec Reordering section on page 17-4.

When a SIP endpoint makes an SDP call, a list of codecs is provided for each media stream in an endpoint. The codecs are listed by payload types in the m= attribute in the order of highest to lowest preference. The SIP endpoint chooses the highest-priority codec that is acceptable to the SIP.

For example, a call uses G.711 and G.723 codec, but uses the G.711 where possible. To set the preference to G.711, the SIP places the PCMU and PCMA before G.723 in its m-line:

\[
m=audio 1234 RTP/AVP 8 0 4.\]

Configuring Reordering

To reorder the codecs in a list, and to prioritize a codec within the list, configure the list of codecs, provide a description for the list, and then add the priority to the codecs in the list.

SUMMARY STEPS

1. configure terminal
2. sbc service-name
3. sbe
4. codec list name
5. description text
6. codec codec-name [packetization-period packet-period [priority priority-value] | priority priority-value [packetization-period packet-period]]
7. end
8. show sbc service-name sbe codec-list list-name

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure terminal</td>
<td>Enables the global configuration mode.</td>
</tr>
</tbody>
</table>

Example:

Router# configure terminal

Step 2 sbc service-name

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Enters the submode of an SBC service.</td>
</tr>
<tr>
<td>Router(config)# sbc mysbc</td>
<td></td>
</tr>
</tbody>
</table>
## Codecs Preference and Reordering Support

### Command or Action			Purpose

<table>
<thead>
<tr>
<th>Step 3</th>
<th>sbe</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc)# sbe</td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>Configures the submode of the SBE entity within an SBC service.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 4</th>
<th>codec list name</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe)# codec list my_codecs</td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>Creates a codec list.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 5</th>
<th>description text</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-codec-list)# description Legitimate codes</td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>Adds a description for the specified codec list using a readable text string format.</td>
</tr>
<tr>
<td></td>
<td>The no form of this command removes the description.</td>
</tr>
<tr>
<td></td>
<td>This description is displayed when the show command is used for this codec list. It is also included for each codec list when a summary of all codec lists is displayed.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 6</th>
<th>codec codec-name <strong>[packetization-period packet-period priority priority-value]</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-codec-list)# codec G723 priority 1</td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>Adds a codec to a codec list, sets a minimum packetization period (optional) for the codec, and sets a priority for the codec that is used for reordering.</td>
</tr>
<tr>
<td></td>
<td>The default codec preference priority is 100.</td>
</tr>
<tr>
<td></td>
<td>A smaller priority value indicates a higher priority, for example if the codec preference value is 1, which is the highest priority and places the codec at the top of the list.</td>
</tr>
<tr>
<td></td>
<td>Using the no form of this command (without the packetization period) removes the named codec from the codec list.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> If the no form of this command includes the packetization period, only the packetization period for the codec is removed.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 7</th>
<th>end</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-codec-list)# end</td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>Exits the codec-list mode and enters the Privileged EXEC mode.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 8</th>
<th>show sbc service-name <strong>sbe codec-list list-name</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Router# show sbc mysbc sbe codec-list my_codecs</td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>Displays detailed information about the codec lists configured on the SBE.</td>
</tr>
<tr>
<td></td>
<td>If the list name for example, my_codecs, is omitted, details are displayed for all the codec lists on the SBE.</td>
</tr>
</tbody>
</table>
Configuring Codec Preferences

To add codec preferences within a list, configure the CAC with a list of codecs, provide a description for the list, and then add the preference to the codecs.

**SUMMARY STEPS**

1. `configure terminal`
2. `sbc service-name`
3. `sbe`
4. `cac-policy-set policy-set-id`
5. `first-cac-scope scope-name`
6. `cac-table table-name`
7. `table-type {policy-set | limit {list of limit tables}}`
8. `entry entry-id`
9. `codec-preference-list list-name`
10. `end`
11. `show`. `sbc service-name sbe cac-policy-set id table name entry entry`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>configure terminal</code></td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>sbc service-name</code></td>
<td>Enters the submode of an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# sbc mysbc</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>sbe</code></td>
<td>Configures the submode of the SBE entity within a SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> <code>cac-policy-set policy-set-id</code></td>
<td>Enters the submode of CAC policy.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe)# cac-policy-set 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> <code>first-cac-table table-name</code></td>
<td>Configures the name of the first policy table to processed when performing the admission control stage of the policy.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-cacpolicy)# first-cac-table first_policy_table</td>
<td></td>
</tr>
<tr>
<td>Step 6</td>
<td>Command or Action</td>
</tr>
<tr>
<td>-------</td>
<td>---------------------------</td>
</tr>
<tr>
<td></td>
<td>cac-table table-name</td>
</tr>
</tbody>
</table>

**Example:**
```
Router(config-sbc-sbe-cacpolicy)# cac-table
first_policy_table
```
Command or Action

**Step 7**

```text
table-type {policy-set | limit {list of limit tables}})
```

**Example:**

```
Router(config-sbc-sbe-cacpolicy-cactable)#
table-type policy-set
```

**Purpose**

Configures the table type of a CAC table within the context of an SBE policy set.

The *list of limit tables* argument controls the syntax of the match-value fields of the entries in the table. Possible available Limit tables are:

- `account`—Compare the name of the account.
- `adj-group`—Compare the name of the adjacency group.
- `adjacency`—Compare the name of the adjacency.
- `all`—No comparison type. All events match this type.
- `call-priority`—Compare with call priority.
- `category`—Compare the number analysis assigned category.
- `dst-account`—Compare the name of the destination account.
- `dst-adj-group`—Compare the name of the destination adjacency group.
- `dst-adjacency`—Compare the name of the destination adjacency.
- `dst-prefix`—Compare the beginning of the dialed digit string.
- `event-type`—Compare with CAC policy event types.
- `src-account`—Compare the name of the source account.
- `src-adj-group`—Compare the name of the source adjacency group.
- `src-adjacency`—Compare the name of the source adjacency.
- `src-prefix`—Compare the beginning of the calling number string.

**Note**

For Limit tables, the event or message or call matches only a single entry.

Features can be enabled or disabled per adjacency group through CAC configuration the same way this is done per individual adjacencies. The adj-group table type matches on either source or destination adjacency group.

When the policy-set keyword is specified, use the **cac-scope** command to configure the scope within each entry at which limits are applied in a CAC Policy Set table.

**Note**

For Policy Set tables, the event or call or message is applied to all entries in this table.
### Configuration Examples—Configuring Codec Restriction

This section provides a sample configuration and output for configuring restrictions on codecs and configuring a CAC policy to use a codec list.

#### Example of Configuring Codecs

The following example shows the commands required to configure codec restriction.

*Figure 17-1* contains three adjacencies (A, B, and C). Any calls involving A need to be configured to use only the G729 and PCMU (G.711) codecs with a minimal preferred packetization period of 10 milliseconds. However, calls between B and C can use any available codecs.

To create a codec list containing the specified codecs configured with a minimal preferred packetization period, use the following commands:

```
Router# configure terminal
Router(config)# sub mysbc
Router(config-sbc)# sbe
```

For more information about the commands, refer to the *Cisco Unified Border Element (SP Edition) Command Reference: Unified Model*.

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 8</strong></td>
<td></td>
</tr>
<tr>
<td><code>entry entry-id</code></td>
<td>Enters the entry mode to create or modify an entry in an admission control table.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-sbc-sbe-cacpolicy-cactable)# entry 1</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td></td>
</tr>
<tr>
<td><code>codec-preference-list list-name</code></td>
<td>Configures the CAC to set preferences for the codecs in the codecs list.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-sbc-sbe-cacpolicy-cactable-entry)# codec-preference-list my_codecs</code></td>
<td>Configures the CAC to set preferences for the codecs in the codecs list. Using the no form of this command, or not setting this command, allows the recognized codecs, if any, to be used without any usage preference. <em>Note</em> This command replaces the codec-preference-list, if any, set up by an earlier CAC entry. To clear all the preferences from an earlier CAC entry, configure a codec-preference-list list-name, where list-name is the name of a list containing no codecs.</td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td></td>
</tr>
<tr>
<td><code>end</code></td>
<td>Exits the cac-policy-set configuration mode and enters the Privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-sbc-sbe-cacpolicy)# end</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 11</strong></td>
<td></td>
</tr>
<tr>
<td><code>show sbc service-name sbe cac-policy-set id table name entry entry</code></td>
<td>Displays detailed information pertaining to a specific entry in a CAC policy table, including codecs with preferences, if any.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router# show sbc mysbc sbe cac-policy-set 1 table standard_policy_list entry 1</code></td>
<td>Displays detailed information pertaining to a specific entry in a CAC policy table, including codecs with preferences, if any.</td>
</tr>
</tbody>
</table>
Chapter 17      Codec Handling

Configuration Examples—Configuring Codec Restriction

Router(config-sbc-sbe)# codec list allowable_codecs
Router(config-sbc-sbe-codec-list)# description The set of codecs allowed on adjacency AdjA
Router(config-sbc-sbe-codec-list)# codec g729 packetization-period 20
Router(config-sbc-sbe-codec-list)# codec pcmu packetization-period 10
Router(config-sbc-sbe-codec-list)# exit

After configuring codec restriction, you must configure a CAC policy to use the codec list. See Example of Configuring a CAC Policy to Use a Codec List section on page 17-23.

Example of Configuring a CAC Policy to Use a Codec List

The following example shows the commands required to configure a CAC policy to use a codec list. To configure a codec list, see Example of Configuring Codecs section on page 17-22.

Router# configure terminal
Router(config)# sbc mysbc
Router(config-sbc)# sbe
Router(config-sbc-sbe)# cac-policy-set 1
Router(config-sbc-sbe-cacpolicy)# first-cac-table table1
Router(config-sbc-sbe-cacpolicy)# first-cac-scope call
Router(config-sbc-sbe-cacpolicy)# cac-table table1
Router(config-sbc-sbe-cacpolicy-cactable)# match-type adjacency
Router(config-sbc-sbe-cacpolicy-cactable)# entry 1
Router(config-sbc-sbe-cacpolicy-cactable)# match AdjA
Router(config-sbc-sbe-cacpolicy-cactable)# codec-restrict-to-list allowable_codecs
Router(config-sbc-sbe-cacpolicy-cactable-entry)# action cac-complete
Router(config-sbc-sbe-cacpolicy-cactable-entry)# exit
Router(config-sbc-sbe-cacpolicy)# complete
The codec list command line interface (CLI) commands can only be entered at the per-call level in theCAC policy tables. If you configure a codec list at any other level the CAC policy set will not activate. However, a log is displayed after you complete the configuration and the policy set is marked as “complete”.

### Configuration Examples—H.323 TCS Codecs Support

The following example configures a codec list called “caller-media-caps-list” and assigns that list to theCAC table “cac-tbl-1” in entry 1 to announce media capabilities on behalf of the SIP caller:

```snippet	race
Router(config)# sbc mySBC
Router(config-sbc)# sbe
Router(config-sbc-sbe)# codec list caller-media-caps-list
Router(config-sbc-sbe-codclist)# codec t38
Router(config-sbc-sbe-cacpolicy)# cac-table cac-tbl-1
Router(config-sbc-sbe-capolicy-cactable)# table-type policy-set
Router(config-sbc-sbe-capolicy-cactable-entry)# entry 1
Router(config-sbc-sbe-capolicy-cactable-entry)# caller-media-caps caller-media-caps-list
```

The following example configures a codec list called “callee-media-caps-list” and assigns that list to theCAC table “cac-tbl-1” in entry 1 to announce media capabilities on behalf of the SIP callee:

```snippet	race
Router(config)# sbc mySBC
Router(config-sbc)# sbe
Router(config-sbc-sbe)# codec list callee-media-caps-list
Router(config-sbc-sbe-codclist)# codec t38
Router(config-sbc-sbe-cacpolicy)# cac-table cac-tbl-1
Router(config-sbc-sbe-capolicy-cactable)# table-type policy-set
Router(config-sbc-sbe-capolicy-cactable-entry)# callee-media-caps callee-media-caps-list
```

The following example configures a codec list called “tcs-extra-caps-list” and assigns that list to theCAC table “cac-tbl-1” in entry 1 to announce extra codecs capability on behalf of the SIP side, whether it is the SIP caller or callee:

```snippet	race
Router(config)# sbc mySBC
Router(config-sbc)# sbe
Router(config-sbc-sbe)# codec list tcs-extra-caps-list
Router(config-sbc-sbe-codclist)# exit
Router(config-sbc-sbe-cacpolicy)# cac-table cac-tbl-1
Router(config-sbc-sbe-capolicy-cactable)# table-type policy-set
Router(config-sbc-sbe-capolicy-cactable-entry)# tcs-extra-codecs tcs-extra-caps-list
```

The following example show command lists the codec list names for caller and callee media capabilities and extra TCS capabilities for entry 1 in a CAC policy table:

```snippet	race
Router# show sbc mySBC sbe cac-policy-set 1 table cac-tbl-1 entry 1
SBC Service mySBC
CAC Policy Set 1
Active policy set: No
Description:
Averaging period: 60 sec
First CAC table:
```

```snippet	race
```
First CAC scope: global

Table name: cac-tbl-1
Description:
Table type: policy-set
Total call setup failures (due to non-media limits): 0

Entry 1
CAC scope:
CAC scope prefix length: 0
Action: Not set
Number of call setup failures (due to non-media limits): 0
Max calls per scope: Unlimited
Max in-call rate: Unlimited
Max reg. per scope: Unlimited
Max channels per scope: Unlimited
Early media: Allowed
Early media timeout: None
Callee Bandwidth-Field: None
Media bypass: Allowed
Renegotiate Strategy: Delta
Max bandwidth per scope: Unlimited

Caller media capabilities: caller-media-caps-list
Callee media capabilities: callee-media-caps-list
Extra TCS capabilities: tcs-extra-caps-list

Configuration Example—Defining a Codec using Dynamic Codec Configuration

The following example shows the commands required to define a custom codec from a codec included in SBC (system codec):

Router# configure terminal
Router(config)# sbc mySBC
Router(config-sbc)# sbe
Router(config-sbc-sbe)# codec custom G726-40-4 id 4
Router(config-sbc-sbe-codec-def)# rate 64000
Router(config-sbc-sbe-codec-def)# packet time 100
Router(config-sbc-sbe-codec-def)# bandwidth 128000
Router(config-sbc-sbe-codec-def)# sample size 4
Router(config-sbc-sbe-codec-def)# channels 16
Router(config-sbc-sbe-codec-def)# max-frames-per-packet 12
Router(config-sbc-sbe-codec-def)# media video
Router(config-sbc-sbe-codec-def)# options transcode
Router(config-sbc-sbe-codec-def)# type sampling
SDP Bandwidth Field Features

Cisco Unified Border Element (SP Edition) supports the Bandwidth Field Interworking and the Option to Use CODEC Instead of Bandwidth-Field for Media Bandwidth Allocation features in the unified model.

In a Session Initiation Protocol (SIP) message exchange, Cisco Unified Border Element (SP Edition) uses the parameters defined in the Session Description Protocol (SDP) bandwidth-fields (b-line) for calculating the media pinhole bandwidth. During SIP message exchange, the SDP may contain both the bandwidth-fields and coder/decoder (CODEC) information. In that case, Cisco Unified Border Element (SP Edition) would use the bandwidth-field value to allocate sufficient bandwidth for the media pinhole.

During deployment, there might be some endpoints for which it would be better to set the media pinhole bandwidth using a CODEC definition in the Session Description Protocol (SDP) messages instead of using b-line.

The Option to Use CODEC Instead of Bandwidth-Field for Media Bandwidth Allocation feature allows you to set a media bandwidth flag in a Call Admission Control (CAC) policy entry to ignore the b-line and use CODEC for calculating the media pinhole bandwidth.

Cisco Unified Border Element (SP Edition) supports Bandwidth Field Interworking by supporting the ability to determine how bandwidth lines are translated in the outbound Session Description Protocol (SDP) sent to the caller and callee with bandwidth line passthrough using Application Specific Maximum (AS) and Transport Independent Application Specific Maximum (TIAS) conversion.

Note

For Cisco IOS XE Release 2.4 and later, these features are supported in the unified model only.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).

For a complete description of the commands used in this chapter, refer to the Cisco Unified Border Element (SP Edition) Command Reference: Unified Model at:


For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

Feature History for SDP Bandwidth Field Features

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 2.4</td>
<td>Option to Use CODEC Instead of Bandwidth-Field for Media Bandwidth Allocation feature was introduced on the Cisco IOS XR, along with the unified model.</td>
</tr>
</tbody>
</table>
Chapter 18      SDP Bandwidth Field Features

This chapter contains the following sections:

- Prerequisites for Implementing SDP Bandwidth Field Features, page 18-2
- Option to Use CODEC Instead of Bandwidth-Field for Media Bandwidth Allocation Feature, page 18-2
- Option to Use CODEC Instead of Bandwidth-Field for Media Bandwidth Allocation Configuration: Example, page 18-7
- Information About Media Bandwidth Policy, page 18-8
- Configuring Media Bandwidth Policy, page 18-9
- Bandwidth Field Interworking Feature, page 18-17
- Bandwidth Field Interworking Configuration: Examples, page 18-22
- Per-Adjacency Codec String Interworking, page 18-23

Prerequisites for Implementing SDP Bandwidth Field Features

The following prerequisite is required to implement SDP Bandwidth Field features:

Before implementing SDP Bandwidth Field features, Cisco Unified Border Element (SP Edition) must already be configured.

Option to Use CODEC Instead of Bandwidth-Field for Media Bandwidth Allocation Feature

The following sections are in the “Option to Use CODEC Instead of Bandwidth-Field for Media Bandwidth Allocation” feature:

- Information About Calculating Bandwidth in SIP Calls, page 18-3
- Configuring Option to Use CODEC Instead of Bandwidth-Field for Media Bandwidth Allocation, page 18-3
- Option to Use CODEC Instead of Bandwidth-Field for Media Bandwidth Allocation Configuration: Example, page 18-7
Information About Calculating Bandwidth in SIP Calls

The SBC analyzes each media stream in a call and calculates the bandwidth required. For SIP calls containing SDP, the SBC looks for any b=CT, b=AS, or b=TIAS lines. If present, these lines are used to calculate the baseline bandwidth required for the media stream.

If these lines are not present, the SBC calculates the baseline bandwidth by inspecting each of the possible CODECs in the stream and calculating the baseline bandwidth based on them. The bandwidth allocated per CODEC is configurable.

The SBC then adjusts the baseline bandwidth to take into account any necessary packetization and Real Time Control Protocol (RTCP) bandwidth overheads.

Certain endpoints do not conform to the bandwidth requirements that the SBC calculates for a media stream, for example:

- Endpoints that start renegotiating the bandwidth for a call can start using additional bandwidth before the renegotiation is complete.
- Endpoints that request an incorrect bandwidth for secure media using the b-line, because they do not take into account the increased payload size required for the encryption.
- Endpoints that transmit data in multiple formats in parallel (such as high and low definition video using different payloads in a single stream) without taking into account all formats when calculating the bandwidth requirements.

To allow interoperation with these endpoints without dropping packets, the SBC allows the per-CODEC bandwidth to be configurable. This allows the SBC administrator to set a suitably large maximum value for CODECs supported by these endpoints. This is sufficient because the endpoints in question use a well known set of CODECs.

However, if the endpoint includes an explicit bandwidth (b-) line, then the SBC uses that to calculate the bandwidth instead of the maximum value. The Option to Use CODEC Instead of Bandwidth-Field for Media Bandwidth Allocation feature uses the `media bandwidth-fields ignore` command to set a media flag in a Call Admission Control (CAC) policy entry to ignore the b-line and use CODEC to calculate the bandwidth.

Configuring Option to Use CODEC Instead of Bandwidth-Field for Media Bandwidth Allocation

This task configures the Option to Use CODEC Instead of Bandwidth-Field for Media Bandwidth Allocation feature.

**SUMMARY STEPS**

1. configure terminal
2. sbc service-name
3. sbe
4. cac-policy-set policy-set-id
5. first-cac-table table-name
6. cac-table table-name
7. table-type {policy-set | limit {list of limit tables}}
### Option to Use CODEC Instead of Bandwidth-Field for Media Bandwidth Allocation Feature

8. **entry entry-id**
9. **cac-scope** {list of scope options}
10. **media bandwidth-fields ignore**
11. **action** [next-table goto-table-name | cac-complete]
12. **exit**
13. **exit**
14. **complete**

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enables global configuration mode.</td>
</tr>
</tbody>
</table>

**Example:**
```
Router# configure
```

<table>
<thead>
<tr>
<th><strong>Step 2</strong> sbc service-name</th>
<th>Enters the mode of an SBC service.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Use the <em>service-name</em> argument to define the name of the service.</td>
</tr>
</tbody>
</table>
```
Router(config)# sbc mysbc
```

<table>
<thead>
<tr>
<th><strong>Step 3</strong> sbe</th>
<th>Enters the mode of an SBE entity within an SBC service.</th>
</tr>
</thead>
</table>

**Example:**
```
Router(config-sbc)# sbe
```

<table>
<thead>
<tr>
<th><strong>Step 4</strong> cac-policy-set policy-set-id</th>
<th>Enters the mode of CAC policy set configuration within an SBE entity, creating a new policy set if necessary.</th>
</tr>
</thead>
</table>

**Example:**
```
Router(config-sbc-sbe)# cac-policy-set 1
```

<table>
<thead>
<tr>
<th><strong>Step 5</strong> first-cac-table table-name</th>
<th>Configures the name of the first policy table to process when performing the admission control stage of policy.</th>
</tr>
</thead>
</table>

**Example:**
```
Router(config-sbc-sbe-cacpolicy)#
first-cac-table StandardListByAccount
```

<table>
<thead>
<tr>
<th><strong>Step 6</strong> cac-table table-name</th>
<th>Enters the mode for configuration of an admission control table (creating one if necessary) within the context of an SBE policy set.</th>
</tr>
</thead>
</table>

**Example:**
```
Router(config-sbc-sbe-cacpolicy)#
cac-table StandardListByAccount
```
### Step 7

**Command or Action**: `table-type {policy-set | limit (list of limit tables)}`

**Example:**

```
Router(config-sbc-sbe-cacpolicy-cactable)#
table-type policy-set
```

**Purpose**: Configures the table type of a CAC table within the context of an SBC policy set. The `list of limit tables` argument controls the syntax of the match-value fields of the entries in the table. Possible available Limit tables are:

- `account`—Compare the name of the account.
- `adj-group`—Compare the name of the adjacency group.
- `adjacency`—Compare the name of the adjacency.
- `all`—No comparison type. All events match this type.
- `call-priority`—Compare with call priority.
- `category`—Compare the number analysis assigned category.
- `dst-account`—Compare the name of the destination account.
- `dst-adj-group`—Compare the name of the destination adjacency group.
- `dst-adjacency`—Compare the name of the destination adjacency.
- `dst-prefix`—Compare the beginning of the dialed digit string.
- `event-type`—Compare with CAC policy event types.
- `src-account`—Compare the name of the source account.
- `src-adj-group`—Compare the name of the source adjacency group.
- `src-adjacency`—Compare the name of the source adjacency.
- `src-prefix`—Compare the beginning of the calling number string.

Features can be enabled or disabled per adjacency group through CAC configuration the same way this is done per individual adjacencies. The `adj-group` table type matches on either source or destination adjacency group.

When the `policy-set` keyword is specified, use the `cac-scope` command to configure the scope within each entry at which limits are applied in a CAC Policy Set table.

### Step 8

**Command or Action**: `entry entry-id`

**Example:**

```
Router(config-sbc-sbe-cacpolicy-cactable)# entry 1
```

**Purpose**: Enters the mode to create or modify an entry in an admission control table.
Option to Use CODEC Instead of Bandwidth-Field for Media Bandwidth Allocation Feature

### Step 9

**Command or Action**

```plaintext
 cac-scope {list of scope options}
```

**Example:**

```plaintext
Router(config-sbc-sbe-cacpolicy-cactable-entry)
# cac-scope src-adjacency
```

**Purpose**

Choose a scope at which CAC limits are applied within each of the entries in a policy set table.

- `list of scope options`—Specifies one of the following strings used to match events:
  - `account`—Events that are from the same account.
  - `adjacency`—Events that are from the same adjacency.
  - `adj-group`—Events that are from members of the same adjacency group.
  - `call`—Scope limits are per single call.
  - `category`—Events that have same category.
  - `dst-account`—Events that are sent to the same account.
  - `dst-adj-group`—Events that are sent to the same adjacency group.
  - `dst-adjacency`—Events that are sent to the same adjacency.
  - `dst-number`—Events that have same destination.
  - `global`—Scope limits are global
  - `src-account`—Events that are from the same account.
  - `src-adj-group`—Events that are from the same adjacency group.
  - `src-adjacency`—Events that are from the same adjacency.
  - `src-number`—Events that have the same source number.
  - `sub-category`—The limits specified in this scope apply to all events sent to or received from members of the same subscriber category.
  - `sub-category-pfx`—The limits specified in this scope apply to all events sent to or received from members of the same subscriber category prefix.
  - `subscriber`—The limits specified in this scope apply to all events sent to or received from individual subscribers (a device that is registered with a Registrar server).

### Step 10

**Command or Action**

```plaintext
 media bandwidth-fields ignore
```

**Example:**

```plaintext
Router(config-sbc-sbe-cacpolicy-cactable-entry)
# media bandwidth-fields ignore
```

**Sets the media flag to ignore the b-line and use CODEC to calculate the baseline bandwidth required for the media stream.**
### Option to Use CODEC Instead of Bandwidth-Field for Media Bandwidth Allocation Configuration: Example

The following example shows how to set the media flag to ignore the b-line and use CODEC to calculate the baseline bandwidth required for the media stream:

```
Router# configure terminal
Router(config)# sbc mySbc
Router(config-sbc)# sbe
Router(config-sbc-sbe)# cac-policy-set 1
Router(config-sbc-sbc-cacpolicy)# first-cac-table StandardListByAccount
Router(config-sbc-sbc-cacpolicy)# cac-table StandardListByAccount
Router(config-sbc-sbc-cacpolicy-cactable)# table-type policy-set
Router(config-sbc-sbc-cacpolicy-cactable)# entry 1
Router(config-sbc-sbc-cacpolicy-cactable-entry)# cac-scope src-adjacency
Router(config-sbc-sbc-cacpolicy-cactable-entry)# media bandwidth-fields ignore
Router(config-sbc-sbc-cacpolicy-cactable-entry)# action cac-complete
Router(config-sbc-sbc-cacpolicy-cactable-entry)# exit
Router(config-sbc-sbc-cacpolicy-cactable)# exit
Router(config-sbc-sbc-cacpolicy)# complete
```

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 11</strong></td>
<td>Configures the action to perform after this entry in an admission control table. Possible actions are:</td>
</tr>
<tr>
<td>**action [next-table goto-table-name</td>
<td>**</td>
</tr>
<tr>
<td><strong>cac-complete]</strong></td>
<td>Stop processing for this scope using the <strong>cac-complete</strong> keyword.</td>
</tr>
</tbody>
</table>
| **Example:**      | Router(config-sbc-sbc-cacpolicy-cactable-entry)
|                   | # action cac-complete |
| **Step 12**       | Exits from **entry** to **cactable** mode. |
| **exit**          | Router(config-sbc-sbc-cacpolicy-cactable-entry)
|                   | # exit |
| **Step 13**       | Exits from **cactable** to **cacpolicy** mode. |
| **exit**          | Router(config-sbc-sbc-cacpolicy-cactable)
|                   | # exit |
| **Step 14**       | Completes the CAC policy set when you have committed the full set. |
| **complete**      | Router(config-sbc-sbc-cacpolicy)
|                   | # complete |
### Information About Media Bandwidth Policy

Previous to this release, SBC disabled or rejected media calls that exceeded the bandwidth allowed by the Call Admission Control (CAC). However, some applications, such as Telepresence, require other options. SBC now provides the ability to diminish the video stream to a lower bandwidth, while allowing the audio stream to remain unchanged.

In this release, bandwidth restrictions are enhanced by allowing the user to configure one of three media bandwidth options, using the `media police` command:

- `strip`
- `reject`
- `degrade`

These options can be configured for all media types or for video only if desired.

#### strip

If an individual media stream exceeds the bandwidth limit for a call, that media stream is disabled by setting the port to zero (0). If after the above stage has completed, the sum of the bandwidths of all remaining streams exceeds the bandwidth limit for a call, the request is rejected.

When the port is set to zero (0), the call is ended and the following message is displayed on the screen:

```
incompatible sites
```

#### reject

If an individual media stream exceeds the bandwidth limit for a call, the request is rejected. If the sum of the bandwidths of all media streams exceeds the bandwidth limit for a call, the request is rejected.

#### degrade

If a media stream exceeds the bandwidth limit for a call, the video stream is downgraded to a lower (non-zero) bandwidth that brings the media stream within the bandwidth limit for the call.

**Note** Only the video stream is downgraded. Audio streams are not downgraded. If the audio stream exceeds the bandwidth for a call, the media stream cannot be downgraded.

### Restrictions

The degrade option is not supported on H.323 calls.

Using the degrade option may cause a 2 to 5 percent performance degradation.

### Configuration

You configure the Media Bandwidth Policy by configuring the media policy mode, using the `media policy` command, and by configuring the minimum bandwidth for the analog-to-digital codec (enCODer/DECoder) hardware, using the `bandwidth` command.

**Note** The codec name must be one of the system codecs that SBC can recognize. To see a list of the system codecs, use the `show sbc sbc sbe codecs` command.

The `max-bandwidth-per-scope` command specifies the maximum bandwidth limit for all media streams in all directions, including packet overheads.

The `bandwidth min` command specifies the unidirectional, minimum bandwidth limit bandwidth and does not include packet overhead.
See the Configuring Media Bandwidth Policy section on page 18-9 for the configuration steps and the end section on page 18-30 for configuration examples.

Configuring Media Bandwidth Policy

This section provides the following step procedures:

- Configuring the Media Policy Mode, page 18-9
- Configuring the Codec Minimum Bandwidth, page 18-10
- Configuring the Maximum Bandwidth Per Scope, page 18-12

Configuring the Media Policy Mode

Use the following procedure to configure the media policy mode.

**SUMMARY STEPS**

1. `config`
2. `sbc sbc-name`
3. `sbe`
4. `cac-policy-set policy-set-id`
5. `cac-table table-name`
6. `table-type policy-set`
7. `entry entry-id`
8. `media police strip | reject | degrade`
9. `end`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><code>config</code></td>
</tr>
</tbody>
</table>
| **Example:** | `config` | Route
| **Step 2** | `sbc sbc-name` | Creates the SBC service on Cisco Unified Border Element (SP Edition) and enters into SBC configuration mode. |
| **Example:** | `sbc SBC1` | Route
| **Step 3** | `sbe` | Enters the mode of the signaling border element (SBE) function of the SBC. |
| **Example:** | `sbe` | Route
### Configuring Media Bandwidth Policy

**SUMMARY STEPS**

1. `config`
2. `sbc sbc-name`
3. `sbe`
4. `codec custom custom-name id`
5. `type variable`
6. `media video`
7. `bandwidth min bandwidth-value`
8. `end`

**EXAMPLE:**

```plaintext```
Example:
```bash```
```shell```
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong>&lt;br&gt;Router# config</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td><code>sbc scl-name</code></td>
<td>Creates the SBC service on Cisco Unified Border Element (SP Edition) and enters into SBC configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong>&lt;br&gt;Router(config)# sbc SBC1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td><code>sbe</code></td>
<td>Enters the mode of the signaling border element (SBE) function of the SBC.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong>&lt;br&gt;Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td><code>codec custom custom-name id</code></td>
<td>Specifies the name of the custom analog-to-digital codec (enCOder/DECoder) and enters codec definition mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong>&lt;br&gt;Router (config-sbc-sbe)# codec custom h263-c id 96</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td><code>type variable</code></td>
<td>Sets the type of the codec to variable.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong>&lt;br&gt;Router (config-sbc-sbe-codec-def)# type variable</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td><code>media video</code></td>
<td>Sets the media type to video.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong>&lt;br&gt;Router (config-sbc-sbe-codec-def)# media video</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td><code>bandwidth min bandwidth-value</code></td>
<td>Sets the minimum bandwidth for the codec. <strong>Note</strong> The <code>bandwidth min</code> command specifies the unidirectional, minimum bandwidth limit and does not include packet overhead.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong>&lt;br&gt;Router (config-sbc-sbe-codec-def)# bandwidth min 328000</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td><code>end</code></td>
<td>Exits configuration mode and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong>&lt;br&gt;Router(config-sbc-sbe-cacpolicy-cactable-entry) # end</td>
<td></td>
</tr>
</tbody>
</table>
Configuring the Maximum Bandwidth Per Scope

Use the following procedure to configure the bandwidth limit for all media streams.

SUMMARY STEPS

1. configure terminal
2. sbc sbc-name
3. sbe
4. cac-policy-set policy-set-id
5. description description
6. first-cac-table table-name
7. first-cac-scope scope-name
8. cac-table table-name
9. table-type policy-set
10. entry entry-id
11. max-bandwidth-per-scope bandwidth
12. action cac-complete
13. media police degrade
14. complete
15. codec system sys-codec id payload id
16. type variable
17. bandwidth min bandwidth-value
18. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td></td>
</tr>
<tr>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# config t</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td></td>
</tr>
<tr>
<td>sbc sbc-name</td>
<td>Creates the SBC service on Cisco Unified Border Element (SP Edition) and enters into SBC configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# sbc SBC1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td></td>
</tr>
<tr>
<td>sbe</td>
<td>Enters the mode of the signaling border element (SBE) function of the SBC.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
</tbody>
</table>
| **Step 4** cac-policy-set policy-set-id | Enters the mode of CAC policy set configuration within an SBE entity, creating a new policy set if necessary.  
*policy-set-id*—Integer chosen by the user to identify the policy set. The range is 1 to 2147483647. |
| **Example:**  
Router(config-sbc-sbe)# cac-policy-set 1 | |
| **Step 5** description description | Configures descriptive text for this policy set. |
| **Example:**  
Router(config-sbc-sbe-cacpolicy)# description bandwidth degrade | |
| **Step 6** first-cac-table table-name | Configures the name of the first policy table to process. A CAC policy may have many tables configured. To start the application of the CAC policy, the first table that is used needs to be defined.  
table-name—The admission control table that should be processed first. |
| **Example:**  
Router(config-sbc-sbe-cacpolicy)# first-cac-table my_table | |
| **Step 7** first-cac-scope scope-name | Configures scope at which limits should be initially defined when performing the admission control stage of the policy. Each CAC policy has a scope that is applied to it. This CAC policy applies on a per call basis.  
*scope-name* has one of the following values: |
| **Example:**  
Router(config-sbc-sbe-cacpolicy)# first-cac-scope call | |
| • adj-group—Limits for events from members of the same adjacency group. |
| • call—Limits are per single call. |
| • category—Limits per category. |
| • dst-account—Limits for events sent to the same account. |
| • dst-adj-group—Limits for events sent to the same adjacency group. |
| • dst-adjacency—Limits for events sent to the same adjacency. |
| • dst-number—Limits for events that have the same adjacency number. |
| • global—Limits are global (May not be combined with any other option). |
| • src-account—Limits for events from the same account. |
| • src-adj-group—Limits for events from the same adjacency group. |
| • src-adjacency—Limits for events from the same adjacency. |
| • src-number—Limits for events that have the same source number. |
Configuring Media Bandwidth Policy

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 8</strong></td>
<td><strong>cac-table</strong> table-name</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-cacpolicy)# cac-table testSecure</td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>Enters the mode for configuration of an admission control table (creating one if necessary) within the context of an SBE policy set.</td>
</tr>
<tr>
<td><strong>table-name</strong>—Name of the admission control table.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td><strong>table-type</strong> (policy-set</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-cacpolicy-cactable)# table-type policy-set</td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>Configures the table type of a CAC table within the context of an SBE policy set. For Policy Set tables, the event or call or message is applied to all entries in this table.</td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td><strong>entry</strong> entry-id</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-cacpolicy-cactable)# entry 1</td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>Enters the mode to modify an entry in an admission control table.</td>
</tr>
<tr>
<td><strong>entry-id</strong>—Specifies the table entry.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 11</strong></td>
<td><strong>max-bandwidth-per-scope</strong> bandwidth</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-cacpolicy-cactable-entry)# max-bandwidth-per-scope 6000000 bps</td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>Configures the maximum limit for the bandwidth in bps, Kbps, Mbps or Gbps for an entry in an admission control table.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>The <strong>max-bandwidth-per-scope</strong> command specifies the maximum bandwidth limit for all media streams in all directions, including packet overheads.</td>
</tr>
<tr>
<td><strong>Step 12</strong></td>
<td><strong>action</strong> cac-complete</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-cacpolicy-cactable-entry)# action cac-complete</td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>Configures the action to perform after this entry in an admission control table. In this case, stop processing for this scope using the <strong>cac-complete</strong> keyword.</td>
</tr>
<tr>
<td><strong>Step 13</strong></td>
<td><strong>media</strong> police strip</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-cacpolicy-cactable-entry)# media police degrade</td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>Configures how SBC handles media streams that exceed bandwidth limits for media calls. In this case, <strong>degrade</strong>.</td>
</tr>
<tr>
<td><strong>Step 14</strong></td>
<td><strong>complete</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-cacpolicy-cactable-entry)# complete</td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>Completes the CAC-policy or call-policy set after committing the full set.</td>
</tr>
<tr>
<td><strong>Step 15</strong></td>
<td><strong>codec</strong> system sys-codec id payload id</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-cacpolicy)# codec system H263 id 34</td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>Enters codec definition mode to modify an existing codec.</td>
</tr>
</tbody>
</table>
### Configuring Media Bandwidth Policy

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 16</strong></td>
<td><strong>type variable</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router (config-sbc-sbe-codec-def)# type variable</td>
</tr>
<tr>
<td>Sets the type of the codec to variable.</td>
<td></td>
</tr>
</tbody>
</table>

| **Step 17** | **bandwidth min bandwidth-value** |
| **Example:** | Router (config-sbc-sbe-codec-def)# bandwidth min 370000 |
| Sets the minimum bandwidth for the codec. |

**Note** The `bandwidth min` command specifies the unidirectional, minimum bandwidth limit bandwidth and does not include packet overhead.

| **Step 18** | **end** |
| **Example:** | Router(config-sbc-sbe-cacpolicy-cactable-entry) # end |
| Exits configuration mode and returns to privileged EXEC mode. |
Media Policy Configuration: Examples

This section provides the following configuration examples:

- Media Policy Mode Configuration: Example, page 18-16
- Codec Minimum Bandwidth Configuration: Example, page 18-16
- Maximum Bandwidth Per Scope Configuration: Example, page 18-16

Media Policy Mode Configuration: Example

The following example shows how to configure SBC to degrade media streams to lower bandwidths when requests exceed bandwidth limits.

Router# config t
Router(config)# sbc mySBC
Router(config-sbc)# sbe
Router(config-sbc-sbe)# cac-policy-set 1
Router(config-sbc-sbe-cacpolicy)# cac-table cac-tbl-1
Router(config-sbc-sbe-cacpolicy-cactable)# table-type policy-set
Router(config-sbc-sbe-cacpolicy-cactable)# entry 1
Router(config-sbc-sbe-cacpolicy-cactable-entry)# media police degrade

Codec Minimum Bandwidth Configuration: Example

The following example shows how to configure the maximum bandwidth limit to 400,000 bps for media calls:

Router# config t
Router(config)# sbc mySBC
Router(config-sbc)# sbe
Router(config-sbc-sbe)# codec system H263 id 34
Router(config-sbc-sbe-codec-def)# bandwidth 400000

The following example shows how to configure the minimum bandwidth limit to 328,000 bps, specifically for video type media calls:

Router# config t
Router(config)# sbc mySBC
Router(config-sbc)# sbe
Router(config-sbc-sbe)# codec custom h263-c id 96
Router(config-sbc-sbe-codec-def)# type variable
Router(config-sbc-sbe-codec-def)# media video
Router(config-sbc-sbe-codec-def)# bandwidth min 328000

Maximum Bandwidth Per Scope Configuration: Example

The following example shows how to configure the bandwidth limit for all media streams:

Router# config t
Router(config)# sbc SBC1
Router(config-sbc)# sbe
Router(config-sbc-sbe)# cac-policy-set 1
Router(config-sbc-sbe-cacpolicy)# description bandwidth degrade
Router(config-sbc-sbe-cacpolicy)# first-cac-table my_table
Router(config-sbc-sbe-cacpolicy)# first-cac-scope call
Bandwidth Field Interworking Feature

The following sections are in the Bandwidth Field Interworking feature:

- Information About Bandwidth Field Interworking, page 18-17
- Configuring Bandwidth Field Interworking, page 18-18
- Bandwidth Field Interworking Configuration: Examples, page 18-22

Information About Bandwidth Field Interworking

Cisco Unified Border Element (SP Edition) supports Bandwidth Field Interworking by supporting the ability to determine how bandwidth lines are configured in the outbound Session Description Protocol (SDP). Cisco Unified Border Element (SP Edition) supports the Application Specific Maximum (AS) and Transport Independent Application Specific Maximum (TIAS) bandwidth modifiers in the SDP.

The SDP includes an optional bandwidth attribute with the following syntax, according to RFC 3556, Session Description Protocol (SDP) Bandwidth Modifiers:

\[ \text{b=} \text{<modifier>:<bandwidth-value>} \]

The <modifier> is an alphanumeric word that indicates the bandwidth to be used by the media or session. The <bandwidth-value> default is kilobits per second.

The AS bandwidth modifier is used to specify the total bandwidth for a single media stream from one source.

The TIAS bandwidth value is the maximum bandwidth required by the SDP session level or media stream without counting IP or other transport layers like TCP or UDP (RFC 3890).

Cisco Unified Border Element (SP Edition) supports translation between the AS and TIAS bandwidth formats which are configured for each adjacency by means of the following commands:

- caller-bandwidth-field [as-to-tias | tias-to-as]
- callee-bandwidth-field [as-to-tias | tias-to-as]

When you configure the bandwidth line to the as-to-tias setting, this causes the SBC, in an outbound SDP offer, to convert a b=AS line into a b=TIAS line. If there are multiple b=AS lines, only the first line is converted into a b=TIAS line and the rest are ignored.

Translating from a bandwidth modifier of AS into TIAS can be useful in the following situations:

- If operating with upstream devices that only support the AS bandwidth modifier, use of the TIAS bandwidth modifier downstream may improve the accuracy of bandwidth calculations in the network. In some network scenarios, use of the AS bandwidth modifier may lead to incorrect bandwidth calculations, for example, if routing between an IPv4 and IPv6 network (see RFC 3890).
Bandwidth Field Interworking Feature

- For interoperability purposes—if there are downstream devices that do not understand the AS bandwidth modifier.

When you configure the bandwidth line to the tias-to-as setting, this causes the SBC, in an outbound SDP offer, to convert a b=TIAS line into a b=AS line if there is not already a b=AS line associated with that SDP media descriptor. If there are multiple b=TIAS lines, only the first is converted into a b=AS line and the rest are ignored.

Translating from a bandwidth modifier of TIAS into AS can be useful in the following situation:
- For interoperability purposes—if there are downstream nodes that do not understand the TIAS bandwidth modifier.

The SBC supports translation between these two formats. If bandwidth line conversion is enabled for the offerer-side adjacency, then an answer has its bandwidth lines converted to the specified format before being sent back to the offerer. Similarly, if bandwidth line conversion is enabled for the answer-side adjacency, then an offer has its bandwidth lines converted to the specified format before being sent to the answer.

The same rules are applied to translation of bandwidth lines in the answer and translation of bandwidth lines in the offer. The rules are as follows:
- The SBC conforms to whichever outgoing bandwidth line format is configured. If the outgoing adjacency is configured to prefer a specific style of bandwidth line format, then that format is used. Thus any AS or TIAS bandwidth lines are translated to that format.
- If the offerer’s adjacency has no configured bandwidth line format preference, but a translation in bandwidth line format was made on the offer to accommodate the answer-side adjacency’s preference, then the reverse translation is done on the answer.

For example, the answer adjacency is configured to translate to TIAS bandwidth lines. The offerer’s adjacency has no preference. The offerer makes an SDP offer containing b=AS lines, which are then converted by the SBC to the b=TIAS style for the outgoing offer.

The answerer responds with a b=TIAS line which represents an increased bandwidth requirement. This increased bandwidth line is translated back to b=AS before being sent to the offerer because that is the style the offerer last offered with.

Configuring Bandwidth Field Interworking

This task configures the Bandwidth Field Interworking feature.

**Note**
The caller and callee commands have been used in this procedure. In some scenarios, the branch command can be used as an alternative to the caller and callee command pair. The branch command has been introduced in Release 3.5.0. See the Configuring Directed Nonlimiting CAC Policies? section on page 7-37 for information about this command.

**SUMMARY STEPS**

1. configure
2. sbc service-name
3. sbe
4. cac-policy-set policy-set-id
5. first-cac-table table-name
6. **cac-table table-name**
7. **table-type {policy-set | limit {list of limit tables}}**
8. **entry entry-id**
9. **cac-scope {list of scope options}**
10. **caller-bandwidth-field [as-to-tias] [tias-to-as]**
11. **callee-bandwidth-field [as-to-tias] [tias-to-as]**
12. **action [next-table goto-table-name | cac-complete]**
13. **exit**
14. **exit**
15. **complete**

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc service-name</td>
<td>Enters the mode of an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# sbc mySbc</td>
<td>• Use the <code>service-name</code> argument to define the name of the service.</td>
</tr>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Enters the mode of an SBE entity within an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> cac-policy-set policy-set-id</td>
<td>Enters the mode of CAC policy set configuration within an SBE entity, creating a new policy set if necessary.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe)# cac-policy-set 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> first-cac-table table-name</td>
<td>Configures the name of the first policy table to process when performing the admission control stage of policy.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-cacpolicy)# first-cac-table StandardListByAccount</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> cac-table table-name</td>
<td>Enters the mode for configuration of an admission control table (creating one if necessary) within the context of an SBE policy set.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-cacpolicy)# cac-table StandardListByAccount</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action | Purpose
---|---
**Step 7**
`table-type (policy-set | limit (list of limit tables))` | Configures the table type of a CAC table within the context of an SBC policy set. The `list of limit tables` argument controls the syntax of the match-value fields of the entries in the table. Possible available Limit tables are:
- `account`—Compare the name of the account.
- `adj-group`—Compare the name of the adjacency group.
- `adjacency`—Compare the name of the adjacency.
- `all`—No comparison type. All events match this type.
- `call-priority`—Compare with call priority.
- `category`—Compare the number analysis assigned category.
- `dst-account`—Compare the name of the destination account.
- `dst-adj-group`—Compare the name of the destination adjacency group.
- `dst-adjacency`—Compare the name of the destination adjacency.
- `dst-prefix`—Compare the beginning of the dialed digit string.
- `event-type`—Compare with CAC policy event types.
- `src-account`—Compare the name of the source account.
- `src-adj-group`—Compare the name of the source adjacency group.
- `src-adjacency`—Compare the name of the source adjacency.
- `src-prefix`—Compare the beginning of the calling number string.

Features can be enabled or disabled per adjacency group through CAC configuration the same way this is done per individual adjacencies. The `adj-group` table type matches on either source or destination adjacency group.

When the policy-set keyword is specified, use the `cac-scope` command to configure the scope within each entry at which limits are applied in a CAC Policy Set table.

**Example:**
`Router(config-sbc-sbe-cacpolicy-cactable)# table-type policy-set`

**Step 8**
`entry entry-id` | Enters the mode to create or modify an entry in an admission control table.

**Example:**
`Router(config-sbc-sbe-cacpolicy-cactable)# entry 1`
Chapter 18      SDP Bandwidth Field Features

**Command or Action**

**Step 9**  
\[ \text{cac-scope \{list of scope options\}} \]

**Example:**

```plaintext
Router(config-sbc-sbe-cacpolicy-cactable-entry)
# cac-scope src-adjacency
```

**Purpose**

Choose a scope at which CAC limits are applied within each entry in a Policy Set table.

*list of scope options*—Specifies one of the following strings used to match events:

- **account**—Events that are from the same account.
- **adjacency**—Events that are from the same adjacency.
- **adj-group**—Events that are from members of the same adjacency group.
- **call**—Scope limits are per single call.
- **category**—Events that have same category.
- **dst-account**—Events that are sent to the same account.
- **dst-adj-group**—Events that are sent to the same adjacency group.
- **dst-adjacency**—Events that are sent to the same adjacency.
- **dst-number**—Events that have the same destination.
- **global**—Scope limits are global
- **src-account**—Events that are from the same account.
- **src-adj-group**—Events that are from the same adjacency group.
- **src-adjacency**—Events that are from the same adjacency.
- **src-number**—Events that have the same source number.

**Step 10**  
\[ \text{caller-bandwidth-field \{as-to-tias\}} \]
\[ \text{\{tias-to-as\}} \]

**Example:**

```plaintext
Router(config-sbc-sbe-cacpolicy-cactable-entry)
# caller-bandwidth-field as-to-tias
```

**Purpose**

Configures the SBC to convert a b=AS line format into a b=TIAS line format or a b=TIAS line format into a b=AS line format in an outbound Session Description Protocol (SDP) sent to the caller.

**Step 11**  
\[ \text{callee-bandwidth-field \{as-to-tias\}} \]
\[ \text{\{tias-to-as\}} \]

**Example:**

```plaintext
Router(config-sbc-sbe-cacpolicy-cactable-entry)
# callee-bandwidth-field tias-to-as
```

**Purpose**

Configures the SBC to convert a b=AS line format into a b=TIAS line format or a b=TIAS line format into a b=AS line format in an outbound Session Description Protocol (SDP) sent to the callee.

AS = Application Specific Maximum

TIAS = Transport Independent Application Specific Maximum
Bandwidth Field Interworking Configuration: Examples

The caller and callee commands have been used in this procedure. In some scenarios, the branch command can be used as an alternative to the caller and callee command pair. The branch command has been introduced in Release 3.5.0. See the Configuring Directed Nonlimiting CAC Policies? section on page 7-37 for information about this command.

The following example shows that the SBC is configured to convert an AS bandwidth line format into a TIAS bandwidth line format on the offerer-side adjacency (caller side), and to convert a TIAS bandwidth line format into an AS bandwidth line format on the answerer-side adjacency (callee side):

```plaintext
Router# configure terminal
Router(config)# sbc mysbc
Router(config-sbc)# sbe
Router(config-sbc-sbe)# cac-policy-set 1
Router(config-sbc-sbe-cacpolicy)# cac-table 1
Router(config-sbc-sbe-cacpolicy-cactable)# entry 1
Router(config-sbc-sbe-cacpolicy-cactable-entry)# caller-bandwidth-field as-to-tias
Router(config-sbc-sbe-cacpolicy-cactable-entry)# callee-bandwidth-field tias-to-as
```

The following example lists detailed information, including caller and callee bandwidth field information, for entry 1 of CAC table 1:

```plaintext
Router# show sbc mysbc sbe cac-policy-set 1 table 1 entry 1
SBC Service "mysbc"
```
CAC Policy Set 1
Active policy set: Yes
Description:
Averaging period: 60 sec
First CAC table: 1
First CAC scope: global

Table name: cacTable
Description:
Table type: policy-set
Total call failures: 0

Entry 1
CAC scope:
Action: CAC complete
Max calls per scope: Unlimited
Max in-call rate: Unlimited
Max reg. per scope: Unlimited
Max channels per scope: Unlimited
Early media: Allowed
Early media timeout: None
Number of calls rejected: 0
Max call rate per scope: Unlimited
Max out-call rate: Unlimited
Max reg. rate per scope: Unlimited
Max updates per scope: Unlimited
Early media direction: Both
Max call rate per scope: Unlimited
Max reg. rate per scope: Unlimited
Max updates per scope: Unlimited
Transcoder per scope: Allowed

Callee Bandwidth-Field: TIAS-to-AS
Caller Bandwidth-Field: AS-to-TIAS
Media bypass: Allowed
Media flag: Ignore bandwidth-fields (b=)
Renegotiate Strategy: Delta
Max bandwidth per scope: Unlimited
SRTP Transport: Trusted-Only (by default)
Caller hold setting: Standard
Callee hold setting: Standard
Caller privacy setting: Never hide
Callee privacy setting: Never hide
Caller voice QoS profile: Default
Caller video QoS profile: Default
Caller sig QoS profile: Default
Callee voice QoS profile: Default
Callee video QoS profile: Default
Callee sig QoS profile: Default
Restrict codecs to list: Default
Restrict caller codecs to list: Default
Restrict callee codecs to list: Default
Caller inbound SDP policy: None
Caller outbound SDP policy: None
Callee inbound SDP policy: None
Callee outbound SDP policy: None

Per-Adjacency Codec String Interworking

The following sections are in the Per-Adjacency Codec String Interworking feature:
- Information about Per-Adjacency Codec String Interworking, page 18-24
- Restrictions for Per-Adjacency Codec String Interworking, page 18-24
- Configuring Per-Adjacency Codec String Interworking, page 18-24
- Configuration Example for Per-Adjacency Codec String Interworking, page 18-30
Information about Per-Adjacency Codec String Interworking

From Cisco IOS Release 3.2S, the SBC can interpret non-standard SDP, and convert codecs between different non-standard forms of SDP or convert non-standard SDP to standard SDP, so that different non-standard devices can interwork through the SBC.

The SBC works through each codec in the message and determines whether the codec name received on the corresponding inbound SDP is a standard name or a variant, and then converts the codec:

- If the codec is standard, SBC searches through the variant list looking for a matching variant of that standard codec. The matching variant, if found, is converted or passed through unchanged.
- If the codec is a variant, SBC searches through the profile to see if that variant is listed.
  - Listed variant passes through unchanged
  - Unlisted variant can get converted to a matching variant of the same standard codec.
  - Unlisted variant that does not have a matching variant of the same standard codec gets converted to the standard representation

In the following scenarios, the codec convert should be avoided and SBC should use the standard SDP form:

- To ensure consistency of the H.248 interface, SDP fragments sent by SBC-SIG in H.248 commands to the media gateways (MGs) must use standard representations. The H.248 interface must be specified to facilitate interoperability with MGs and make it easy for the third party MGs to implement the SBC H.248 profile.
- To ensure consistency of the billing interface, the SDP fragments stored by SBC-SIG in the XML billing records must use standard representations. It is easy for the third party billing servers to parse XML billing records.
- Codec strings do not appear in H.245 signaling messages, an enumerated type is used to represent the codec. Therefore, the codec convert is only applicable for SIP outbound adjacency.

Restrictions for Per-Adjacency Codec String Interworking

The Per-Adjacency Codec String Interworking feature has the following restrictions:

- When a particular variant is used for a given codec on passing through an Offer, the same variant may not be used when passing through the Answer.
- For a given side of the call, SBC cannot be configured to interpret the incoming SDP using one variant but convert outgoing SDP based on another.
- You cannot define a variant that uses a standard IANA codec string because the SBC supports only those variants that use non-standard strings.
- If two variants map to the same standard codec, the transcoder does not convert between them. For example, the SBC cannot transcode between G7231H and G7231L, even though the endpoints can perceive those to be different codecs.

Configuring Per-Adjacency Codec String Interworking

This section explains the following configurations for Per-Adjacency Codec String Interworking feature:

- Configuring Codec Variant Conversion, page 18-25
Configuring Codec Variant Conversion

This task explains how to configure codec variant conversion on the SBC:

**SUMMARY STEPS**

1. configure terminal
2. sbc service-name
3. sbe
4. codec system sys-codec id payload-id
5. fmtp fmtp-string
6. exit
7. codec variant codec variant-name
8. variant variant-codec-encoded-name
9. standard standard-codec-name
10. fmtp fmtp-string
11. exit
12. codec variant profile profile-name
13. variant variant-name
14. end
15. show sbc service-name sbe codecs variant [profile]

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>configure terminal</td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>sbc service-name</td>
<td>Enters the SBC service mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>Use the service-name argument to define the name of the service.</td>
</tr>
<tr>
<td>Step 3</td>
<td>sbe</td>
<td>Enters the SBE entity mode within an SBC service.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
</tbody>
</table>

Example:
```
Router# configure
```

Example:
```
Router(config)# sbc mysbc
```

Example:
```
Router(config-sbc)# sbe
```
### Per-Adjacency Codec String Interworking

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 4</strong></td>
<td><strong>codec system sys-codec id payload-id</strong></td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe)# codec system G723 id 4</td>
</tr>
<tr>
<td></td>
<td>Enters the codec definition mode to modify an existing codec.</td>
</tr>
<tr>
<td></td>
<td>• <em>sys-codec</em>—The codec included in the SBC.</td>
</tr>
<tr>
<td></td>
<td>• <em>id payload-id</em>—Static payload id. Value can be from 0 to 96.</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td><strong>fmtp fmtp-string</strong></td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-codec-def)# fmtp annexa=yes</td>
</tr>
<tr>
<td></td>
<td>Configures the default value of Format-Specific Parameters (FMTP) for SDP.</td>
</tr>
<tr>
<td></td>
<td>• <em>fmtp-string</em>—The FMTP string for SDP, in the name=value format.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>To view the default FMTP values associated with variants, use the <code>show sbc sbe codecs variant</code> command.</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td><strong>exit</strong></td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-codec-def)# exit</td>
</tr>
<tr>
<td></td>
<td>Exits from the codec definition mode and enters into the SBE entity mode.</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td><strong>codec variant codec variant-name</strong></td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe)# codec variant codec G723-H-1</td>
</tr>
<tr>
<td></td>
<td>Enters the codec variant mode to configure, modify, or delete a codec variant.</td>
</tr>
<tr>
<td></td>
<td>• <em>variant-name</em>—The codec variant name.</td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td><strong>variant variant-codec-encoded-name</strong></td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-codec-var-codec)# variant G723-H-1</td>
</tr>
<tr>
<td></td>
<td>Defines the encoded codec variant name.</td>
</tr>
<tr>
<td></td>
<td>• <em>variant-codec-encoded-name</em>—The variant nonstandard codec string.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>‘#’ is reserved for base variants. Therefore, the variant name cannot start with ‘#’</td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td><strong>standard standard-codec-name</strong></td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-codec-var-codec)# standard G723</td>
</tr>
<tr>
<td></td>
<td>Defines the standard codec variant name.</td>
</tr>
<tr>
<td></td>
<td>• <em>standard-codec-name</em>—The standard system codec name.</td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td><strong>fmtp fmtp-string</strong></td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-codec-var-codec)# fmtp bitrate=6.3</td>
</tr>
<tr>
<td></td>
<td>Define the FMTP parameters for the codec variant.</td>
</tr>
<tr>
<td></td>
<td>• <em>fmtp-string</em>—The FMTP string in the name=value format.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>To view the default FMTP values associated with variants, use the <code>show sbc sbe codecs variant</code> command.</td>
</tr>
<tr>
<td><strong>Step 11</strong></td>
<td><strong>exit</strong></td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-codec-var-codec)# exit</td>
</tr>
<tr>
<td></td>
<td>Exits the codec variant mode and enters into the SBE entity mode.</td>
</tr>
</tbody>
</table>
Chapter 18  SDP Bandwidth Field Features

Per-Adjacency Codec String Interworking

Chapter 18      SDP Bandwidth Field Features

Configuring Codec on CAC Policy Set

This task shows how to enable codec convert and configure codec variant profile on a CAC policy set. When the codec variant conversion is enabled or disabled, the following events occur:

- If the codec variant conversion is disabled, the SBC does not take into account the specified variant profile. All the codecs that have been passed are left in their original representation, and any new codecs added by the SBC are added with their standard representation.
- If the codec variant conversion is enabled but the variant profile is not configured, all the codecs are converted to its standard representation.
- If the codec variant is enabled and the variant profile is configured, any codecs matched by the profile are converted to the appropriate variant representation, and any codecs that is not matched by the variant profile is converted to its standard representation.

Note

The caller and callee commands have been used in this procedure. In some scenarios, the branch command can be used as an alternative to the caller and callee command pair. The branch command has been introduced in Release 3.5.0. See the ?paranum>Configuring Directed Nonlimiting CAC Policies? section on page 7-37 for information about this command.

SUMMARY STEPS

1. configure terminal
2. sbc service-name
3. sbe
4. cac-policy-set policy-set-id
5. cac-table table-name
6. `table-type {policy-set | limit [list of limit tables]}`
7. `entry entry-id`
8. `caller codec convert`
9. `callee codec convert`
10. `caller codec profile profile-name`
11. `callee codec profile profile-name`
12. `exit`
13. `exit`
14. `complete`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>configure terminal</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>Router# configure</code></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><code>sbc service-name</code></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>Router(config)# sbc mysbc</code></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><code>sbe</code></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>Router(config-sbc)# sbe</code></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td><code>cac-policy-set policy-set-id</code></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>Router(config-sbc-sbe)# cac-policy-set 1</code></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td><code>cac-table table-name</code></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>Router(config-sbc-sbe-cacpolicy)# cac-table StandardListByAccount</code></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>`table-type {policy-set</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>Router(config-sbc-sbe-cacpolicy-cactable)# table-type policy-set</code></td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
</tr>
<tr>
<td>------</td>
<td>-------------------</td>
</tr>
<tr>
<td>7</td>
<td>entry entry-id</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-cacpolicy-cactable)# entry 1</td>
</tr>
<tr>
<td>8</td>
<td>caller codec convert</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-cacpolicy-cactable-entry) # caller codec convert</td>
</tr>
<tr>
<td>9</td>
<td>callee codec convert</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-cacpolicy-cactable-entry) # callee codec convert</td>
</tr>
</tbody>
</table>
| 10   | caller codec profile profile-name | To specify a codec variant profile at the caller side.  
|      | **Example:**      |         |
|      | Router(config-sbc-sbe-cacpolicy-cactable-entry) # caller codec profile |         |
| 11   | callee codec profile profile-name | To specify a codec variant profile at the callee side.  
|      | **Example:**      |         |
|      | Router(config-sbc-sbe-cacpolicy-cactable-entry) # callee codec profile |         |
| 12   | exit              | Exits from the entry mode and enters into the cactable mode. |
|      | **Example:**      |         |
|      | Router(config-sbc-sbe-cacpolicy-cactable-entry) # exit |         |
| 13   | exit              | Exits from the cactable mode and enters into the cacpolicy mode. |
|      | **Example:**      |         |
|      | Router(config-sbc-sbe-cacpolicy-cactable)# exit |         |
| 14   | complete          | Completes the CAC policy set when you have committed the full set. |
|      | **Example:**      |         |
|      | Router(config-sbc-sbe-cacpolicy)# complete |         |
Configuration Example for Per-Adjacency Codec String Interworking

The following example shows how to configure the Per-Adjacency Codec String Interworking feature when caller supports G723 bitrate 6.3 annexa=no codec and callee supports G.723.1 codec variant:

```plaintext
configure terminal
sbc MySBC
sbe
codec variant codec PCMU.1
    standard PCMU
    variant PCMU.1
    exit
codec variant profile pcmu-var
    variant PCMU.1
    variant #G.723.1/H
    exit
cac-policy-set 2
    first-cac-table codec-convert
    first-cac-scope src-adjacency
    cac-table codec-convert
    table-type limit src-adjacency
    entry 1
        match-value CallMgrA
        callee codec profile pcmu-var
        callee codec convert
        media police strip
        action cac-complete
        complete
        end
```
SDP Handling

Cisco Unified Border Element (SP Edition) by default passes through all a= lines in SIP messages containing SDP offers and answers that it forwards. You can also configure Cisco Unified Border Element (SP Edition) to block certain a= lines, either by specifying a whitelist (a finite set of a=lines that are passed through, with all others blocked), or alternatively a blacklist (a finite set of a=lines that are blocked, with all others passed through). Additionally, user exits in the Cisco Unified Border Element (SP Edition) code base allow customers to write their own code to insert and/or strip one or more media-level a= lines when processing an offer on an answer.

The SIP-I Support feature enables Cisco Unified Border Element (SP Edition) to pass through the ISDN User Part (ISUP) parameters in Session Initiation Protocol (SIP) messages that are added by a SIP or Public Switched Telephone Network (PSTN) interworking gateway.

The SIP Non-SDP Body Filtering feature adds support for Cisco Unified Border Element (SP Edition) to process non-SDP bodies, and in particular the ISUP body using SIP-I.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).

For a complete description of the commands used in this chapter, refer to the Cisco Unified Border Element (SP Edition) Command Reference: Unified Model at:


For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

Feature History for SDP Handling

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 2.4</td>
<td>The SIP SDP Attribute Passthrough feature was introduced on the Cisco IOS XR.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 2.6</td>
<td>The SIP-I Support and SIP Non-SDP Body Filtering features were introduced on the Cisco ASR 1000 Series Aggregation Services Routers.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 3.1S</td>
<td>Repeat SDP on 200 Invite Response feature was added.</td>
</tr>
</tbody>
</table>
Contents

This chapter contains the following sections:

- Configuring SIP SDP Attribute Passthrough, page 19-2
- SIP-I Support and SIP Non-SDP Body Filtering, page 19-17

Configuring SIP SDP Attribute Passthrough

This section contains the following subsections:

- Restrictions for Configuring SIP SDP Attribute Passthrough, page 19-2
- Information about SIP SDP Attribute Passthrough, page 19-3
- Information About Repeat SDP on 200 Invite Response, page 19-3
- Configuring SIP SDP Attribute Passthrough, page 19-6
- Configuring Repeat SDP on 200 INVITE Response, page 19-14
- Example of SIP SDP Attribute Passthrough, page 19-15
- Example of Repeat SDP on 200 INVITE Response Configuration, page 19-16

Restrictions for Configuring SIP SDP Attribute Passthrough

Review the following restrictions for SIP SDP Attribute Passthrough:

- The existing reflect behavior is not supported.
- Wildcard or prefix matching of attribute lines is not supported.
- Distinguishing media-level from session-level a-lines for the purposes of matching is not supported.
- Sophisticated matching conditions (for example, apply only to video streams or apply only to offers) are not supported.
- Attribute blocking in media bypass calls is not supported.
- Blocking function is restricted to unknown attributes.
- The following attributes are ignored by unknown attribute policy because this may interfere with the correct operation of the SBC:
  - a=rtpmap
  - a=fmtp
  - a=sendonly
  - a=recvonly
  - a=inactive
  - a=sendrecv
  - a=ptime
  - a=mid
  - a=group
  - a=curr
- `a=des`
- `a=conf`
- `a=crypto`.

At the point where the policy is applied, a (rate-limited) warning log is issued if the policy attempts to delete one of these lines.

### Information about SIP SDP Attribute Passthrough

Additional per-call storage is needed to store the SDP policy that is being applied. This is expected to be ~160 bytes per call.

### Information About Repeat SDP on 200 Invite Response

To support interoperability with endpoints that may require an agreed Session Description Protocol (SDP) to be resent for 200 INVITE responses, the user can configure SBC to repeat an agreed SDP, in a 200 INVITE response, when needed, after the successful provisioning of an offer-answer exchange.

This option is configured in the CAC policy for SIP calls. The default is off.

The agreed SDP answer is the SDP answer from the latest completed SDP offer/answer exchange procedure.

When Repeat SDP on 200 Invite Response is configured, the call flow is as shown in the following three figures.
Figure 19-1 shows the call flow for an SDP on the second reliable response.

**Figure 19-1 Call Flow for SDP on Second Reliable Response**

```
Caller  

INVITE (SDP offer)

183 Session Progress 100 rel (SDP answer)

PRACK

200 PRACK

183 Session Progress 100 rel (No SDP)

PRACK

200 PRACK

200 OK (repeat SDP answer)

SBC  

INVITE (SDP offer)

183 Session Progress 100 rel (SDP answer)

PRACK

200 PRACK

183 Session Progress 100 rel (SDP)

SBC ignores SDP in second 183 msg

PRACK

200 PRACK

200 OK

SBC ignores SDP in 200 OK

SBC repeat SDP in 200 OK

Callee
```
Figure 19-2 shows the call flow for an SDP on the final response.

**Figure 19-2  Call Flow for SDP on Final Response**

- **Caller**
  - INVITE (SDP offer)
  - 183 Session Progress 100 rel (SDP answer)
  - PRACK
  - 200 PRACK
  - 200 OK (SDP Answer)
  - ACK

- **SBC**
  - INVITE (SDP offer)
  - 183 Session Progress 100 rel (SDP answer)
  - PRACK
  - 200 PRACK
  - SBC repeat SDP in 200OK

- **Callee**
  - 200 OK (New SDP)
  - ACK
Figure 19-3 shows the call flow for no SDP on the final response.

Figure 19-3  Call Flow for No SDP on Final Response

See the ‘Configuring Repeat SDP on 200 INVITE Response?’ section on page 19-14 for the configuration procedure.

See the ‘Example of Repeat SDP on 200 INVITE Response Configuration?’ section on page 19-16 for the example configuration.

Configuring SIP SDP Attribute Passthrough

This section contains the steps for implementing SIP SDP Attribute Passthrough.

Note

The caller and callee commands have been used in this procedure. In some scenarios, the branch command can be used as an alternative to the caller and callee command pair. The branch command has been introduced in Release 3.5.0. See the ‘Configuring Directed Nonlimiting CAC Policies?’ section on page 7-37 for information about this command.
SUMMARY STEPS

1. configure
2. sbc service-name
3. sbe
4. sip sdpmatch-table table-name1
5. action whitelist | blacklist
6. match-string attribute-name1
7. match-string attribute-name2
8. exit
9. sip sdpmatch-table table-name2
10. action whitelist | blacklist
11. match-string attribute-name1
12. match-string attribute-name3
13. exit
14. sip sdp-policy-table table-name1
15. match-table table-name 1
16. exit
17. sip sdp-policy-table table-name2
18. match-table table-name2
19. exit
20. cac-policy-set number
21. first-cac-table table-name
22. first-cac-scope scope
23. cac-table table-name
24. table-type {policy-set | limit {list of limit tables}}
25. entry number
26. match-value value
27. action action-name
28. caller-inbound-policy policytab-name
29. caller-outbound-policy policytab-name
30. callee-inbound-policy policytab-name
31. callee-outbound-policy policytab-name
32. exit
33. exit
34. complete
35. exit
36. active-cac-policy-set number
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc service-name</td>
<td>Enters the mode of an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# sbc mysbc</td>
<td></td>
</tr>
<tr>
<td>- Use the <code>service-name</code> argument to define the name of the service.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Enters the mode of the signaling border element (SBE) function of the SBC.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> sip sdp-match-table table-name</td>
<td>Adds an existing sdp-match-table into policy.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe)# sip sdp-match-table 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> action whitelist/blacklist</td>
<td>Specifies an SDP policy table action.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-sdp-match-tbl)# action blacklist</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> match-string attribute-name1</td>
<td>Configures an SDP attribute matching string.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-sdp-match-tbl)# match-string X-sqn1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> match-string attribute-name1</td>
<td>Configures an SDP attribute matching string.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-sdp-match-tbl)# match-string X-sqn2</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> exit</td>
<td>Returns to the previous submode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-sdp-match-tbl)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong> sip sdp-match-table table-name</td>
<td>Adds an existing sdp-match-table into policy.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe)# sip sdp-match-table 2</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 10</strong> action whitelist/blacklist</td>
<td>Adds an action allowing a defined set of attributes and blocking the remaining attributes.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-sdp-match-tbl)# action blacklist</td>
<td></td>
</tr>
<tr>
<td><strong>Step 11</strong> match-string attribute-name1</td>
<td>Configures an SDP attribute matching string.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-sdp-match-tbl)# match-string X-sqn1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 12</strong> match-string attribute-name1</td>
<td>Configures an SDP attribute matching string.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-sdp-match-tbl)# match-string X-sqn2</td>
<td></td>
</tr>
<tr>
<td><strong>Step 13</strong> exit</td>
<td>Returns to the previous submode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-sdp-match-tbl)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 14</strong> sip sdp-policy-table table-name</td>
<td>Configures an SDP policy table.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-sip)# sip sdp-policy-table foo</td>
<td></td>
</tr>
<tr>
<td><strong>Step 15</strong> match-table table-name</td>
<td>Configure an SDP match table used in a policy.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-sdp-policy-tbl)# match-table matchtab2</td>
<td></td>
</tr>
<tr>
<td><strong>Step 16</strong> exit</td>
<td>Returns to the previous submode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-sip-adj)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 17</strong> sip sdp-policy-table table-name</td>
<td>Configures an SDP policy table.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe)# sip sdp-policy-table foo2</td>
<td></td>
</tr>
<tr>
<td><strong>Step 18</strong> match-table table-name</td>
<td>Configure an SDP match table used in a policy.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-sdp-policy-tbl)# match-table matchtab3</td>
<td></td>
</tr>
<tr>
<td><strong>Step 19</strong> exit</td>
<td>Returns to the previous submode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-sdp-policy-tbl)# exit</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td><code>cac-policy-set number</code></td>
<td>Enters the submode of CAC policy set configuration.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe)# cac-policy-set 1</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td><code>first-cac-table table-name</code></td>
<td>Configures the name of the first policy table to process when performing the admission control stage of policy.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-cacpolicy)# first-cac-table RootCacTable</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td><code>first-cac-scope scope</code></td>
<td>Configures the scope at which to begin defining limits when performing the admission control stage of policy.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-cacpolicy)# first-cac-scope src-adjacency</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td><code>cac-table table-name</code></td>
<td>Creates or configures an admission control table.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-cacpolicy)# cac-table RootCacTable</td>
<td></td>
</tr>
</tbody>
</table>
Chapter 19  SDP Handling

19-11

Configuring SIP SDP Attribute Passthrough

Step 24

**table-type** *(policy-set | limit)* *(list of limit tables)*

**Example:**

Router(config-sbc-sbe-cacpolicy-cactable)#

table-type limit call-priority

Configures the table type of a CAC table within the context of an SBE policy set.

*list of limit tables* can be one of the following values:

- **account**—Compare the name of the account.
- **adj-group**—Compare the name of the adjacency group.
- **adacency**—Compare the name of the adjacency.
- **all**—No comparison type. All events match this type.
- **call-priority**—Compare with call priority.
- **category**—Compare the number analysis assigned category.
- **dst-account**—Compare the name of the destination account.
- **dst-adj-group**—Compare the name of the destination adjacency group.
- **dst-adjacency**—Compare the name of the destination adjacency.
- **dst-prefix**—Compare the beginning of the dialed digit string.
- **event-type**—Compare with CAC policy event types.
- **src-account**—Compare the name of the source account.
- **src-adj-group**—Compare the name of the source adjacency group.
- **src-adjacency**—Compare the name of the source adjacency.
- **src-prefix**—Compare the beginning of the calling number string.

Features can be enabled or disabled per adjacency group through CAC configuration the same way this is done per individual adjacencies. The adj-group table type matches on either source or destination adjacency group.

Step 25

**entry** *number*

**Example:**

Router(config-sbc-sbe-cacpolicy-cactable)# entry 1

Creates or modifies an entry in a table.
### Command or Action

<table>
<thead>
<tr>
<th>Step 26</th>
<th>match-value key</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-cacpolicy-cactable-entry)# match-value immediate</td>
</tr>
</tbody>
</table>

**Purpose:**
Configures the match-value of an entry in a CAC Limit table. It is only relevant for Limit table types.

The *key* argument is a string or a keyword based on the table type. The format of the key is determined by the Limit table type (for example, Limit event-type tables or Limit call-priority tables).

For Limit event-type tables (**table-type limit event-type**), the match value string options are the following:

- *call-update*—Compare the beginning of the calling number string.
- *endpoint-reg*—Compare the name of the destination adjacency.
- *new-call*—Compare the beginning of the dialed digit string.

For Limit call-priority tables (**table-type limit call-priority**), the match value string options are the following:

- *critical*—Match calls with resource priority 'critical'.
- *flash*—Match calls with resource priority 'flash'.
- *flash-override*—Match calls with resource priority 'flash-override'.
- *immediate*—Match calls with resource priority 'immediate'.
- *priority*—Match calls with resource priority 'priority'.
- *routine*—Match calls with resource priority 'routine'.

For all other Limit tables, enter a name or digit string **WORD**—Name or digit string to match. (Max Size 255).

<table>
<thead>
<tr>
<th>Step 27</th>
<th>action action-name</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-cacpolicy-cactable-entry)# action cac-complete</td>
</tr>
</tbody>
</table>

**Purpose:**
Specifies the action to take if this entry is chosen.

<table>
<thead>
<tr>
<th>Step 28</th>
<th>caller-inbound-policy policytab-name</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-cacpolicy-cactable-entry)# caller-inbound-policy policytab1</td>
</tr>
</tbody>
</table>

**Purpose:**
Configures a caller inbound SDP policy table.
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 29</strong> caller-outbound-policy policytab-name</td>
<td>Configures a caller outbound SDP policy table.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-cacpolicy-cactable-entry)# caller-outbound-policy policytab1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 30</strong> callee-inbound-policy policytab-name</td>
<td>Configures a callee inbound SDP policy table.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-cacpolicy-cactable-entry)# callee-inbound-policy policytab2</td>
<td></td>
</tr>
<tr>
<td><strong>Step 31</strong> callee-outbound-policy policytab-name</td>
<td>Configures a callee outbound SDP policy table.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-cacpolicy-cactable-entry)# callee-outbound-policy policytab2</td>
<td></td>
</tr>
<tr>
<td><strong>Step 32</strong> exit</td>
<td>Returns to the previous submode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-cacpolicy-cactable-entry)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 33</strong> exit</td>
<td>Returns to the previous submode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-cacpolicy-cactable)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 34</strong> complete</td>
<td>Performs a consistency check on the CAC policy set.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-cacpolicy)# complete</td>
<td></td>
</tr>
<tr>
<td><strong>Step 35</strong> exit</td>
<td>Returns to the previous submode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-cacpolicy)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 36</strong> active-cac-policy-set number</td>
<td>Enters the active CAC policy set.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe)# active-cac-policy-set 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 37</strong> end</td>
<td>Exits SBE mode and enters Privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe)# end</td>
<td></td>
</tr>
<tr>
<td><strong>Step 38</strong> show sbc service-name sbe cac-policy-set number table number entry number</td>
<td>Displays detailed information for a given entry in a CAC policy table.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# do show sbc interwork sbe cac-policy-set 1 table 1 entry 1</td>
<td></td>
</tr>
</tbody>
</table>
Configuring Repeat SDP on 200 INVITE Response

Use the following procedure to configure SBC to send a repeat SDP on 200 INVITE responses.

**SUMMARY STEPS**

1. config
2. sbc sbc-name
3. sbe
4. cac-policy-set policy-set-id
5. cac-table table-name
6. table-type policy-set
7. entry entry-id
8. sdp repeat answer
9. end

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 config</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# config</td>
<td></td>
</tr>
<tr>
<td>Step 2 sbc sbc-name</td>
<td>Creates the SBC service on Cisco Unified Border Element (SP Edition) and enters into SBC configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config)# sbc SBC1</td>
<td></td>
</tr>
<tr>
<td>Step 3 sbe</td>
<td>Enters the mode of the signaling border element (SBE) function of the SBC.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td>Step 4 cac-policy-set policy-set-id</td>
<td>Enters the mode of CAC policy set configuration within an SBE entity, creating a new policy set if necessary.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe)# cac-policy-set 1</td>
<td>policy-set-id—Integer chosen by the user to identify the policy set. The range is 1 to 2147483647.</td>
</tr>
<tr>
<td>Step 5 cac-table table-name</td>
<td>Enters the mode for configuration of an admission control table (creating one if necessary) within the context of an SBE policy set.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-cacpolicy)# cac-table testSecure</td>
<td>table-name—Name of the admission control table.</td>
</tr>
</tbody>
</table>
Chapter 19      SDP Handling

Configuring SIP SDP Attribute Passthrough

This section provides a sample configuration and output for SIP SDP Attribute Passthrough.

Command or Action  Purpose

Step 6  table-type policy-set
Example:
Router(config-sbc-sbe-cacpolicy-cactable)#
table-type policy-set

Step 7  entry entry-id
Example:
Router(config-sbc-sbe-cacpolicy-cactable)#
entry 1

Step 8  sdp repeat answer
Example:
Router(config-sbc-sbe-cacpolicy-cactable-entry)
# sdp repeat answer

Step 9  end
Example:
Router(config-sbc-sbe-cacpolicy-cactable-entry)
# end

Table or Action  Purpose

<table>
<thead>
<tr>
<th>Step 6</th>
<th>table-type policy-set</th>
<th>Configures the table type of a CAC table within the context of an SBE policy set.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-cacpolicy-cactable)# table-type policy-set</td>
<td></td>
</tr>
<tr>
<td>Step 7</td>
<td>entry entry-id</td>
<td>Enters the mode to modify an entry in an admission control table.</td>
</tr>
<tr>
<td>entry-id—Specifies the table entry.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-cacpolicy-cactable)# entry 1</td>
<td></td>
</tr>
<tr>
<td>Step 8</td>
<td>sdp repeat answer</td>
<td>Configures SBC to repeat an agreed Session Description Protocol (SDP), in a 200 INVITE response, after the successful provisioning of an offer-answer exchange when needed.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-cacpolicy-cactable-entry)# sdp repeat answer</td>
<td></td>
</tr>
<tr>
<td>Step 9</td>
<td>end</td>
<td>Exits configuration mode and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-cacpolicy-cactable-entry)# end</td>
<td></td>
</tr>
</tbody>
</table>

Example of SIP SDP Attribute Passthrough

The caller and callee commands have been used in this procedure. In some scenarios, the branch command can be used as an alternative to the caller and callee command pair. The branch command has been introduced in Release 3.5.0. See the ?paranum>Configuring Directed Nonlimiting CAC Policies? section on page 7-37 for information about this command.

Router# config t
Enter configuration commands, one per line. End with CTRL/Z.
Router(config)# sbc interwork
Router(config-sbc)# sbe
Router(config-sbc-sbe)# sip sdp-match-table matchtab1
Router(config-sbc-sbe-sdp-match-tbl)# action blacklist
Router(config-sbc-sbe-sdp-match-tbl)# match-string X-sqn
Router(config-sbc-sbe-sdp-match-tbl)# match-string X-cap
Router(config-sbc-sbe-sdp-match-tbl)# exit
Router(config-sbc-sbe)# sip sdp-match-table matchtab2
Router(config-sbc-sbe-sdp-match-tbl)# action blacklist
Router(config-sbc-sbe-sdp-match-tbl)# match-string X-sqn
Router(config-sbc-sbe-sdp-match-tbl)# match-string X-pc-csuites-rtp
Router(config-sbc-sbe-sdp-match-tbl)# exit
Router(config-sbc-sbe)# sdp-policy-table policytab1
Router(config-sbc-sbe-sdp-policy-tbl)# match-table matchtab1
Router(config-sbc-sbe-sdp-policy-tbl)# exit
Router(config-sbc-sbe)# sip sdp-policy-table policytab2
Router(config-sbc-sbe-sdp-policy-tbl)# match-table matchtab2
Router(config-sbc-sbe-sdp-policy-tbl)# exit
Router(config-sbc-sbe)# cac-policy-set 1
Configuring SIP SDP Attribute Passthrough

This section provides a sample configuration and output for SIP SDP Attribute Passthrough.

Example of Repeat SDP on 200 INVITE Response Configuration

The following example shows how to configure SBC to send a repeat SDP on 200 INVITE responses.
SIP-I Support and SIP Non-SDP Body Filtering

This section contains the following subsections:

- Prerequisites, page 19-17
- Restrictions for SIP Non-SDP Body Filtering and SIP-I Support, page 19-17
- Information about SIP Non-SDP Body Filtering and SIP-I Support, page 19-18
- Configuring SIP SDP Attribute Passthrough, page 19-6
- Examples—SIP Non-SDP Body Filtering and SIP-I Support, page 19-22

Prerequisites

The following prerequisite is required to implement SIP Non-SDP Body Filtering and SIP-I Support:

Before implementing SIP Non-SDP Body Filtering and SIP-I Support, Cisco Unified Border Element (SP Edition) must already be configured. See the procedures described in Chapter 3, ?$paratext>.?

Restrictions for SIP Non-SDP Body Filtering and SIP-I Support

The following restrictions and limitations apply to SIP Non-SDP Body Filtering and SIP-I Support:

- If dual tone multifrequency (DTMF) interworking is enabled for a call, the INFO messages containing a DTMF digit may not pass through.
- The SBC does not support Secure Multipurpose Internet Mail Extensions (S/MIME) encryption or decryption. While the SBC may allow encrypted bodies to pass through, it does not modify them.
- In compliance with Section 8.2.1.1 of RFC 3398, the SBC does not support a From header without a username.
- The total size of the MIME bodies and associated header allowed to pass through is limited to approximately 1000 bytes. The final size allowed depends on the structure of the headers and MIME bodies and should not exceed 2000 bytes.
- The SBC may not preserve the original order of MIME bodies and may insert the SDP as the first body part.
- This feature does not work in conjunction with H.323.
- Since the SBC considers BYE requests on a hop-by-hop basis, it does not pass any information using a BYE response it received.
- The SBC allows the user=phone URI parameter on the Request-URI to pass through.
- The SBC may alter the MIME boundary of a message.
Information about SIP Non-SDP Body Filtering and SIP-I Support

The following sections provide information about the SIP Non-SDP Body Filtering feature and the SIP-I Support feature.

SIP Non-SDP Body Filtering

The SIP Non-SDP Body Filtering feature adds support for Cisco Unified Border Element (SP Edition) to process non-SDP bodies, and in particular the ISUP body using SIP-I. The SBC can pass through, strip out, or reject non-SDP bodies. The message body of a SIP message is described using header fields such as Content-Disposition, Content-Encoding, and ContentType, which provide information about the body. SBC uses a body profile that you create and associate to filter non-SDP bodies from incoming and outgoing SIP messages, based on the Content-Type header field. A body profile allows a message containing a specific non-SDP body to take one of the following actions:

- To be passed (without altering the message)
- To be stripped of the body (and pass the rest of the message)
- To be rejected
- SBC uses the ‘handling’ parameter in the message to decide whether to strip the body or reject the message.

Like any other SIP profile, such as a method profile, you need to first create a body profile. Then you can associate the body profile to cause the body profile to take action on incoming and outgoing SIP messages that fall under the SBE mode, or adjacency mode, or method profile mode.

You can create a body profile:

- Using the `sip body-profile {profile_name}` command under the SBE mode.

  The `body` command and `action (body)` command are used in conjunction with the `sip body-profile` command. The `body` command names a body type or content header type for a non-SDP message body. The `action (body)` command sets the action to take on a body type in a SIP body profile.

After creating a body profile, you can associate the body profile at the following levels and configuration modes:

- At the SIP signaling entity level (ingress or egress), under SBE mode, using the `sip default body-profile [[inbound | outbound] {profile_name}]` command. The body profile is associated for the entire signaling instance (that is all messages, either ingress or egress, passing through the SBC.)
- At the SIP adjacency level, under SIP adjacency mode, using the `body-profile [[inbound | outbound] {profile_name}]` command. The body profile is associated to an adjacency.
- At the SIP method profile level, under method profile mode, using the `body-profile {profile_name}` command. The body profile is associated to a method profile.

SIP-I Support

The SIP-I Support feature enables Cisco Unified Border Element (SP Edition) to pass through the ISDN User Part (ISUP) parameters in Session Initiation Protocol (SIP) messages that are added by a SIP or Public Switched Telephone Network (PSTN) interworking gateway. ISUP is a call control protocol used in SS7 networks primarily for setting up and tearing down telephone calls and for maintenance of the network.
SIP-I is an approach defined by ITU-T Q.1912.5. SIP-I provides an approach for interworking SIP networks and the traditional, circuit-based ISDN User Part (ISUP) networks. SIP-I provides a method for passing through ISUP-specific header parameters through a SIP network so that calls that originate and terminate on the circuit-based ISUP network can cross a SIP network with no loss of information.

SIP-I allows transparent passthrough of ISUP parameters through a SIP network by attaching a copy of the ISUP message to the SIP message at the incoming PSTN gateway. The ISUP message appears as a non-SDP message body on the SIP message. SIP-I has a mechanism to indicate the presence of ISUP (based on Content-Type header) and if the ISUP is mandatory or can be passed through, depending on the Content-Disposition header. The SBC passes through the ISUP message body without coding or decoding the ISUP message.

The mapping between SIP and ISUP protocols is carried out by the Media Gateway Controller (MGC). In the SBC, the ISUP parameters can be carried in the SIP Request-Uniform Resource Identifier (URI) or the SIP message body.

Cisco Unified Border Element (SP Edition) supports the following SIP-I and profile functions:

- Application or SDP is processed on INVITE, UPDATE, and PRACK requests and their responses.
- Application or DTMF-info is processed on INFO to allow DTMF tones to pass through.
- The NOTIFY messages on message or SIP-frag is analyzed to find out whether it indicates that a subscription or refer dialog is to be terminated.

Non-SDP Message Body Example

The following is an example of a non-SDP message body; the SIP message body is not shown in detail for brevity’s sake. The non-SDP body present in the example is of type “application/resource-lists+xml”:

```
INVITE sip:conf-fact@example.com SIP/2.0
   Content-Type: multipart/mixed;boundary="boundary1"
   Content-Length: 617
--boundary1
   Content-Type: application/sdp
   v=0
   o=alice 2890844526 2890842807 IN IP4 atlanta.example.com
   s=-
   c=IN IP4 192.0.2.1
   t=0 0
   m=audio 20000 RTP/AVP 0
   a=rtpmap:0 PCMU/8000
   m=video 20002 RTP/AVP 31
   a=rtpmap:31 H261/90000
--boundary1
   Content-Type: application/resource-lists+xml
   Content-Disposition: recipient-list
   <?xml version="1.0" encoding="UTF-8"?>
   <resource-lists xmlns="urn:ietf:params:xml:ns:resource-lists"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="urn:ietf:params:xml:ns:resource-lists"
      xsi:noNamespaceSchemaLocation="urn:ietf:params:xml:ns:resource-lists"
   >
     <list>
       <entry uri="sip:bill@example.com"/>
       <entry uri="sip:randy@example.net"/>
       <entry uri="sip:joe@example.org"/>
     </list>
   </resource-lists>
--boundary1--
```
Implementing SIP Non-SDP Body Filtering

The follow steps describe a sample configuration where a body profile is created with a particular body type and action to take on that body type and then the body profile is associated at the SIP signaling level.

**SUMMARY STEPS**

1. `configure terminal`
2. `sbc service-name`
3. `sbe`
4. `sip body-profile {profile_name}`
5. `body {WORD}`
6. `action [pass | nopass | strip | reject]`
7. `exit`
8. `exit`
9. `sip default body-profile [[inbound | outbound] {profile_name}]`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>Router# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc service-name</td>
<td>Enters the mode of an SBC service.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>Router(config)# sbc mySBC</code></td>
<td>Use the <code>service-name</code> argument to define the name of the service.</td>
</tr>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Enters the mode of the signaling border element (SBE) function of the SBC.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>Router(config-sbc)# sbe</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> sip body-profile {profile_name}</td>
<td>Creates a body profile to filter non-SDP bodies from incoming and outgoing SIP messages. Enters SBE SIP Body configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>Router(config-sbc-sbe)# sip body-profile bodyprofile1</code></td>
<td></td>
</tr>
</tbody>
</table>
## SDP Handling

### SIP-I Support and SIP Non-SDP Body Filtering

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| 5    | **body** (WORD)   | This command describes the body type or content header type for SBC to act on messages with the specified body type or content header type. Enters SBE SIP Body Element configuration mode.  
Example:  
Router(config-sbc-sbe-sip-body)# body application/ISUP |
| 6    | **action** [pass / nopass / strip / reject] | Sets the action to take on a body type in a SIP body profile for a non-SDP message body.  
- **pass**—Instructs the SBC to pass through the body type of the non-SDP message body.  
- **nopass**—Uses the handling parameter in the message to determine whether to strip the body or reject the entire message with error code 415 (Unsupported media type).  
- **strip**—Strips the body and passes the rest of the message.  
- **reject**—Rejects the entire message with an error code.  
Example:  
Router(config-sbc-sbe-sip-body-ele)# action strip |
| 7    | **exit**          | Exits SBE SIP Body Element configuration mode.  
Example:  
Router(config-sbc-sbe-sip-body-ele)# exit |
| 8    | **exit**          | Exits SBE SIP Body configuration and enters SBE configuration mode.  
Example:  
Router(config-sbc-sbe-sip-body)# exit |
| 9    | **sip default body-profile** [[inbound | outbound] (profile_name)] | Associates the body profile at the SIP signaling level, for the entire signaling instance (that is all messages, either ingress or egress, passing through the SBC).  
Example:  
Router(config-sbc-sbe)# sip default body-profile inbound bodyprofile1 |
Examples—SIP Non-SDP Body Filtering and SIP-I Support

The following is a configuration example of SIP Non-SDP Body Filtering:

```
sbc foo
   sbe
      sip body-profile profile1
         body application/ISUP
            action strip
         body application/QSIG
            action reject
            hunt-on-reject
         body *
            action reject

   sip body-profile profile2
      description test-profile
      body application/ISUP
         action nopass
      body application/QSIG
         action pass

   sip body-profile profile3
      body application/ISUP
         action nopass
      body application/QSIG
         action pass

   sip default body-profile inbound profile1
   sip default body-profile outbound profile2

   sip method-profile default
      !- pre-provisioned
      ! "default" method profile
      !- used at sbe level
      method INVITE
         body-profile profile1

   sip method-profile mp1
      !-create a new method
      ! profile used in adj
      method INVITE
         body-profile profile2

   sip method-profile mp2
      !-create a new method
      ! profile used in adj
      method REGISTER
         body-profile profile1

   adjacency sip adj-1
      body-profile inbound profile2
      body-profile outbound profile1
```

The following example displays all the non-SDP message body profiles in use:

```
Router# show sbc mySBC sbe sip body-profile

<table>
<thead>
<tr>
<th>Name</th>
<th>In Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>profile1</td>
<td>Yes</td>
</tr>
<tr>
<td>profile2</td>
<td>Yes</td>
</tr>
<tr>
<td>profile3</td>
<td>No</td>
</tr>
</tbody>
</table>
```
The following example displays the details of the specified non-SDP message body profile named “profile2”:

```
Router# show sbc mySBC sbe sip body-profile profile2
```

<table>
<thead>
<tr>
<th>Name</th>
<th>profile2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>test-profile</td>
</tr>
<tr>
<td>Element</td>
<td>application/ISUP</td>
</tr>
<tr>
<td>Action</td>
<td>nopass</td>
</tr>
<tr>
<td>Hunt-on-reject</td>
<td>false</td>
</tr>
<tr>
<td>Element</td>
<td>application/QSIG</td>
</tr>
<tr>
<td>Action</td>
<td>pass</td>
</tr>
<tr>
<td>Hunt-on-reject</td>
<td>false</td>
</tr>
</tbody>
</table>
Flexible Media Routing

The Flexible Media Routing feature supports the call legs in which media and signaling are sent over different virtual or physical networks. The signaling network is configured using the vrf command in the adjacency submode. All the call legs to and from an adjacency use the same VPN ID for signaling. The media network is configured using the vrf parameter in the media-address command.

When the Flexible Media Routing feature is enabled, the media address selection overrides the VPN ID-based selection. Therefore, the media VPN ID is no longer compared with the signaling VPN ID. Instead, the SBC selects the media address whose realm matches the adjacency realm. The IPv6 and H.323 protocols support the Flexible Media Routing feature.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to as the session border controller (SBC) in this document.

For a complete description of the commands used in this chapter, refer to the Cisco Unified Border Element (SP Edition) Command Reference: Unified Model at:

For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

Feature History for Flexible Media Routing

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 3.5S</td>
<td>The Flexible Media Routing feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers.</td>
</tr>
</tbody>
</table>

Contents

This chapter contains the following sections:

- Prerequisites for Configuring the Flexible Media Routing Feature, page 20-2
- Configuring the Flexible Media Routing Feature, page 20-2
- Configuration Examples for the Flexible Media Routing Feature, page 20-3
- Changes in XML Billing Records, page 20-4
Prerequisites for Configuring the Flexible Media Routing Feature

The following prerequisite is required to configure the Flexible Media Routing feature:
Ensure that the SBC is deactivated before configuring the Flexible Media Routing feature.

Configuring the Flexible Media Routing Feature

This task shows how to configure the Flexible Media Routing feature.

SUMMARY STEPS

1. configure terminal
2. sbc sbc-name
3. no activate
4. allow diff-med-sig-vpn
5. activate
6. exit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure terminal</td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 2 sbc sbc-name</td>
<td>Enters the SBC configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config)# sbc test</td>
<td></td>
</tr>
<tr>
<td>Step 3 no activate</td>
<td>Deactivates the SBC.</td>
</tr>
<tr>
<td>Example:</td>
<td>Note</td>
</tr>
<tr>
<td>Router(config-sbc)# no activate</td>
<td></td>
</tr>
<tr>
<td>Step 4 allow diff-med-sig-vpn</td>
<td>Allows media and signaling to use different VPN IDs in a call leg.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc)# allow diff-med-sig-vpn</td>
<td></td>
</tr>
</tbody>
</table>

Note: If the SBC is active and you run the allow diff-med-sig-vpn command, the system issues a warning message, asking you to first deactivate the SBC. You can reactivate the SBC using the activate command.

The no version of this command allows media and signaling to use the same VPN ID in a call leg.
The following example shows the SBC behavior when the Flexible Media Routing feature is configured:

```
sbc test
allow diff-med-sig-VPN
```

The following example shows how to configure different VRFs under the signaling and media networks.

```
sbe
adjacency sip sipp1 =============== incoming
  force-signaling-peer all
  vrf vrf_sipp1
  nat force-on
  inherit profile preset-access
  signaling-address ipv4 192.0.2.1
  statistics method summary
  signaling-port 5060
  remote-address ipv4 192.0.2.3 255.255.255.0
  signaling-peer 192.0.2.3
  realm FMR
  attach
adjacency sip sipp2 =============== outgoing
  force-signaling-peer all
  vrf vrf_sipp2
  nat force-off
  inherit profile preset-access
  signaling-address ipv4 192.0.2.2
  statistics method summary
  signaling-port 5060
  remote-address ipv4 192.0.2.4 255.255.255.0
  signaling-peer 192.0.2.4
  fast-register disable
  realm FMR
call-policy-set 1
  first-call-routing-table start-table1
  first-reg-routing-table start-table1
  rtg-src-adjacency-table start-table1
  entry 1
  match-adjacency sipp1
dst-adjacency sipp2
  action complete
  entry 2
  match-adjacency sipp2
dst-adjacency sipp1
  action complete
  complete
call-policy-set default 1
```
network-id 9737
!
media-address ipv4 192.0.2.5 vrf vrf_media realm FMR
  port-range 16384 32767 any
activate
!
end

Changes in XML Billing Records

After the Flexible Media Routing feature is enabled, the SBC adds the mediarealm attribute to the adjacency element in the XML billing records as follows:

<adjacency type="orig" name="Adj1" account="Acc1" vpn="0X12345678" mediarealm="Internet"/>

For more information about the mediarealm attribute, see Appendix C, XML Billing Schema.
Inherit Profiles for Non-IMS Adjacencies

Cisco Unified Border Element (SP Edition) supports Inherit Profiles for adjacencies that are not part of an IP Multimedia Subsystem (IMS) network. This feature allows Cisco Unified Border Element (SP Edition) to operate in non-IMS networks using any of three non-IMS profiles that define an adjacency as Access, Core, or Peering. Cisco Unified Border Element (SP Edition) uses this definition to process packets properly and add the correct information in the outgoing packets.

By configuring each of these different types of adjacency with a profile, you can make efficiency and occupancy gains. For example, Cisco Unified Border Element (SP Edition) does not store registration information from messages received from Peering adjacencies. When a subscriber successfully registers from an Access adjacency, Cisco Unified Border Element (SP Edition) remembers the subscriber's registration details for later use and only stores this information on Access adjacencies.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).

For a complete description of the commands used in this chapter, refer to the Cisco Unified Border Element (SP Edition) Command Reference: Unified Model at:


For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

Note

For Cisco IOS XE Release 2.4, this feature is supported in the unified model only.

Feature History for Inherit Profiles for Non-IMS Adjacencies

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 2.4</td>
<td>This feature was introduced on the Cisco IOS XR.</td>
</tr>
</tbody>
</table>

Contents

This module contains the following sections:

- Prerequisites, page 21-2
- Information About Inherit Profiles for Non-IMS Adjacencies, page 21-2
- CLI Support for Inherit Profiles for Non-IMS Adjacencies, page 21-3
- Configuration Examples, page 21-4
Prerequisites

The following prerequisites are required to implement Inherit Profiles for Non-IMS Adjacencies:

- Before implementing this feature, the SBC must already be configured. See the procedures described in ?Sparatext[CT_ChapTitle]?

Information About Inherit Profiles for Non-IMS Adjacencies

Cisco Unified Border Element (SP Edition) can be deployed in various network topologies and plays different roles depending on its location in the network. Each of the deployed roles usually has a specific set of requirements associated with it. These requirements control which headers need to be added, checked, updated, or removed, and which headers, methods, and options are permitted to be passed through.

Cisco Unified Border Element (SP Edition) can be deployed in non-IMS networks and thus takes on different roles in non-IMS networks. For example, Cisco Unified Border Element (SP Edition) can face a registrar network or end user client devices that will attempt to register through the SBC. Alternatively, you can position it on the Network-Network Interface (NNI).

To deploy in non-IMS networks, Cisco Unified Border Element (SP Edition) uses easily-configured inherit profiles that comprise a collection of related configuration appropriate to a particular network role. Inherit profiles may be configured for an application on a per-adjacency basis or at a global level as a default.

Non-IMS Inherit Profiles Types and Behaviors

The following are the non-IMS inherit profiles that can be configured for an adjacency:

- preset-access profile—configures an Access adjacency. The Access adjacency is not part of an IMS network. This adjacency faces user equipment, such as a subscriber’s telephone or other SIP device, that attempts to register through the SBC.

- preset-core profile—configures a Core adjacency. This is the default profile. The Core adjacency is not part of an IMS network. This adjacency faces a registrar network and links to the registrar.

- preset-peering profile—configures a Peering adjacency. The Peering adjacency is not part of an IMS network. This adjacency, for example, sitting at the Network-Network Interface, links one registrar to another. The SBC is not required to store subscriber information from messages received from peering adjacencies.

The following are examples of behaviors that are affected by the non-IMS inherit profiles:

- Whether various headers (such as P Charging Vector) are created.

- Which headers, methods, and options are passed through and which are stripped out.

- Whether inbound and outbound calls to a subscriber can be made before that subscriber is registered.

- Whether the SBC rewrites the contact headers during the registration process.

When you configure the SBC with a certain non-IMS profile, calls may be handled differently. For example, when a call is received on a Core adjacency, the SBC checks to see if the endpoint is registered. If the subscriber is registered and is known to be behind a Network Address Translation (NAT), the SBC configures the call to traverse the NAT. If the endpoint is not registered, the SBC applies a routing policy and routes the call to the appropriate adjacency.
Effect of Non-IMS Inherit Profiles on Method Profiles, Header Profiles, and Option Profiles

Use of a non-IMS inherit profile dynamically assigns the following sets of profiles (method profile, header profile, and option profile) to a call based on the non-IMS inherit-profile selected. Table 21-1 shows which non-IMS inherit profile has an effect on which specific method profile, header profile, and option profile.

The effect is not visible in the adjacency configuration for header-profile, method-profile or option profiles, and can be overridden by explicit configuration of header, method, option profiles as needed.

Table 21-1 Effect of Non-IMS Inherit Profiles on Method, Header and Option Profiles

<table>
<thead>
<tr>
<th>Non-IMS Inherit Profile</th>
<th>Method Profile</th>
<th>Header Profile</th>
<th>Option Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td>preset-access</td>
<td>preset-std-in-mth</td>
<td>preset-std-in-hdr</td>
<td>preset-std-in-opt</td>
</tr>
<tr>
<td></td>
<td>preset-std-out-mth</td>
<td>preset-std-out-hdr</td>
<td>preset-std-out-opt</td>
</tr>
<tr>
<td></td>
<td>Type: Whitelist</td>
<td>Actions: Passes INFO, Passes UPDATE</td>
<td>Type: Whitelist</td>
</tr>
<tr>
<td></td>
<td>Actions: Passes INFO, Passes UPDATE</td>
<td>Actions: Passes Server, Passes Diversion, Passes Resource-Priority</td>
<td>Actions: Passes Replaces (only)</td>
</tr>
<tr>
<td></td>
<td>preset-std-out-mth</td>
<td>preset-std-out-hdr</td>
<td>preset-std-out-opt</td>
</tr>
<tr>
<td></td>
<td>Type: Whitelist</td>
<td>Actions: Passes Server, Passes Diversion, Passes Resource-Priority</td>
<td>Type: Whitelist</td>
</tr>
<tr>
<td></td>
<td>Actions: Passes INFO, Passes UPDATE</td>
<td>Actions: Passes Server, Passes Diversion, Passes Resource-Priority</td>
<td>Actions: Passes Replaces (only)</td>
</tr>
<tr>
<td></td>
<td>preset-std-out-mth</td>
<td>preset-std-out-hdr</td>
<td>preset-std-out-opt</td>
</tr>
<tr>
<td></td>
<td>Type: Whitelist</td>
<td>Actions: Passes Server, Passes Diversion, Passes Resource-Priority</td>
<td>Type: Whitelist</td>
</tr>
<tr>
<td></td>
<td>Actions: Passes INFO, Passes UPDATE</td>
<td>Actions: Passes Server, Passes Diversion, Passes Resource-Priority</td>
<td>Actions: Passes Replaces (only)</td>
</tr>
</tbody>
</table>

CLI Support for Inherit Profiles for Non-IMS Adjacencies

The inherit profile command has the following three keywords that allow you to configure a preset-access, preset-core, or preset-peer profile for an adjacency that is not part of an IMS network:

- preset-access—Specifies a preset access profile for an adjacency that faces an access device on a User-Network Interface (UNI) location.
- preset-core—Specifies a preset core profile for an adjacency that faces a core device on a UNI location. This is the default.
- preset-peering—Specifies a preset peering profile for an adjacency that faces a peer device on a Network-Network Interface (NNI) location.

The adjacency-specific command configuration overrides any global configuration of the adjacency that was configured using the sip inherit profile command.
The following example shows all the profiles available with the **inherit profile** command:

```
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency sip sipa
Router(config-sbc-sbe-adj-sip)# inherit profile ?

  preset-access  preset-access profile
  preset-core    preset-core profile
  preset-ibcf-ext-untrusted  preset-ibcf-ext-untrusted profile
  preset-ibcf-external      preset-ibcf-external profile
  preset-ibcf-internal      preset-ibcf-internal profile
  preset-p-cscf-access     preset-p-cscf-access profile
  preset-p-cscf-core       preset-p-cscf-core profile
  preset-peering          preset-peering profile
  preset-standard-non-ims  preset-standard-non-ims profile
```

### Configuration Examples

The following example displays detailed output for adjacency client, including the “Inherit profile:” field that shows that the adjacency has been configured with the non-IMS preset-access profile:

```
Router# show sbc mySBC sbe adjacencies client detail

SBC Service "mySBC"
Adjacency client (SIP)
  Status: Attached
  Signaling address: 200.0.0.12:5062, VRF Admin
  Signaling-peer: 200.0.0.30:5062
  Remote address: 200.0.0.0 255.255.255.0
  Force next hop: No
  Account: None
  Group: None
  In header profile: Default
  Out header profile: Default
  In method profile: Default
  Out method profile: Default
  In UA option prof: Default
  Out UA option prof: Default
  In proxy opt prof: Default
  Out proxy opt prof: Default
  Priority set name: None
  Local-id: None
  Rewrite REGISTER: On
  Target address: None
  NAT Status: Auto Detect
  Reg-min-expiry: 3000 seconds
  Fast-register: Enabled
  Fast-register-int: 30 seconds
  Authenticated mode: None
  Authenticated realm: None
  Auth. nonce life time: 300 seconds
  IMS visited NetID: None
  Inherit profile: preset-access
  Force next hop: No
  Home network Id: None
  UnEncrypt key data: None
  SIPI passthrough: No
  Rewrite from domain: Yes
  Rewrite to header: Yes
  Media passthrough: No
  Preferred transport: UDP
```
Hunting Triggers:       Global Triggers
Redirect mode:         Pass-through
Security:              Untrusted
Outbound-flood-rate:   None
Ping-enabled:          No
Signaling Peer Status: Not Tested
Cisco Unified Border Element (SP Edition) Registration Features

Cisco Unified Border Element (SP Edition) supports the SIP Fast Registration, SoftSwitch Shielding, Registration Monitoring, Aggregate Registration, Provisioned Delegate Registration, and Contact Username Passthrough features in the unified model.

SIP Fast Registration addresses the problem where SIP messages to the Network Address Translation (NAT) endpoint are unable to penetrate the NAT and firewalls to establish calls. Using SIP Fast Registration, the NAT endpoints transmit SIP REGISTER requests at a high enough frequency to keep the NAT pinhole alive.

The SoftSwitch Shielding feature allows a lower SIP registration rate on the links to registrars (typically softswitches) than on the links to endpoints. Allowing a lower registration rate shields the softswitch from an undesirably high rate of re-registrations.

Cisco Unified Border Element (SP Edition) supports monitoring events subscription for changes of the registration state with the Registration Monitoring functionality.

Aggregate Registration registers all the endpoints connected to it in a single registration. This functionality enables Cisco Unified Border Element (SP Edition) to support devices that implicitly register multiple endpoints through it.

The Provisioned Delegate Registration feature allows the Cisco Unified Border Element (SP Edition) to support client or end user devices that cannot register themselves in a network where SIP calls are passing through a registrar. Cisco Unified Border Element (SP Edition) is able to register on behalf of such client devices. The Provisioned Delegate Registration feature can support Cisco Telepresence systems where the end user applications cannot send the registration message and Cisco Unified Border Element (SP Edition) does it on their behalf.

The Contact Username Passthrough enhancement enables interoperability with softswitches that require the contact username portion of the Contact URI in SIP REGISTER requests to pass through unchanged.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).

For a complete description of the commands used in this chapter, refer to the Cisco Unified Border Element (SP Edition) Command Reference: Unified Model at:


For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

Note

These features are supported in the unified model.
Chapter 22  Cisco Unified Border Element (SP Edition) Registration Features

Contents

Feature History for Registration Features

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 2.4</td>
<td>The SIP Fast Registration, SoftSwitch Shielding, Registration Monitoring,</td>
</tr>
<tr>
<td></td>
<td>Aggregate Registration, and Delegate Registration features were introduced</td>
</tr>
<tr>
<td></td>
<td>on the Cisco IOS XR along with support for the unified model.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 2.5</td>
<td>The Contact Username Passthrough for non-IMS networks and Support for</td>
</tr>
<tr>
<td></td>
<td>Supported Path Under REGISTER Request features were added.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 3.1S</td>
<td>The Per Subscriber Delete feature was added.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 3.2S</td>
<td>The adding expires-header to register-message feature was added.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 3.3S</td>
<td>The Alternative Contact Rewriting feature was added.</td>
</tr>
</tbody>
</table>

This chapter contains the following sections:

- Prerequisites, page 22-3
- Restrictions, page 22-3
- Information About SIP Registration, page 22-3
- Adding an Expires-Header to a Register-Message, page 22-4
- Support for Supported Path Under REGISTER Request, page 22-6
- Information About Contact Username Passthrough, page 22-6
- Configuring Contact Username Passthrough, page 22-7
- Information About Alternative Contact Rewriting, page 22-9
- Configuring Alternative Contact Rewriting, page 22-10
- Information About SIP Fast Registration, page 22-12
- Configuring SIP Fast Registration, page 22-15
- Information About SoftSwitch Shielding, page 22-16
- Configuring SoftSwitch Shielding, page 22-17
- Information About Registration Monitoring, page 22-19
- Configuring Registration Monitoring, page 22-20
- Information About Per Subscriber Delete, page 22-21
- Configuring Per Subscriber Delete, page 22-21
- Information About Aggregate Registration, page 22-22
- Configuring Aggregate Registration, page 22-22
- Information About Provisioned Delegate Registration, page 22-24
- Provisional Delegate Registration Commands, page 22-26
- Configuration Examples, page 22-30
Prerequisites

The following prerequisite is required to implement SoftSwitch Shielding, Registration Monitoring, Aggregate Registration, and Provisioned Delegate Registration:

Before implementing these features, Cisco Unified Border Element (SP Edition) must already be configured.

Restrictions

SIP Fast Registration has the following restrictions:

- Only UDP is supported.
- REGISTERs with a zero expiry time (“Unregisters”) are always forwarded to the registrar and not fast-pathed, if the SBC matches them to a known registration.
- Minimal parsing of REGISTER requests is performed before a decision is taken to send a fast-path response; this minimizes the load on the SBC. A REGISTER request is only fast-pathed if its expiry interval is not zero and it comes from the same IP address and port as a known subscription.
- Endpoints that send their requests from ephemeral (short-lived) ports do not have their registration requests fast-pathed.
- The “FastReg interval” cannot be higher than the “MinExpiry interval.” If the “MinExpiry interval” is less than twice the “FastReg interval,” fast-pathing is not performed.

Provisioned Delegate Registration has the following restrictions:

- The delegate registration configuration is limited to no more than 1000 subscribers with each subscriber having no more than 5 contacts.
- H.323 adjacencies and SIP to H.323 interworking are not supported in Cisco IOS XE Release 2.4 and earlier.

Information About SIP Registration

Registration is required if the end user has a dynamic IP address, if the provider does not support static hostnames, or if NAT is used.

In a SIP REGISTER message, the Contact: header contains the URI that identifies the subscriber.

When a device registers in a non-IMS network, Cisco Unified Border Element (SP Edition) takes the SIP REGISTER Contact: header and modifies it by replacing the contact username with a hash to produce a unique contact username. This is the default behavior for a typical registration. This is needed because there may be multiple UNI adjacencies in different VLANS, which have similar contacts. (Note that this does not apply to the IMS P-CSCF profile.) Then, the SBC forwards the REGISTER to the registrar containing this new modified Contact: header. Meanwhile, the SBC will also store a record of the original contact and the modified contact in its internal memory.

When the core network desires to ring that subscriber, the SBC will receive an INVITE containing the modified contact information. The Cisco SBC will check its memory to look up the information, and will swap out the header with the original information, and will direct the call to the appropriate SIP adjacency in the correct customer network.

This means that no explicit call routing detail needs to be configured in the call policy for routing calls from the core to subscribers, since the SBC has its internal memory of registrations.
Note that even though a SIP adjacency may be intended to receive only subscriber (registered) traffic, it is still possible for unregistered callers to initiate calls from that same adjacency. This can be considered useful, because emergency callers therefore may not need to register first.

When the call arrives at the softswitch, it can check if the subscriber has registered or not, and if to allow the call or not.

In the case where softswitch interoperability is desired, you may want to pass through the contact username instead of hashing it. Cisco Unified Border Element (SP Edition) provides a Contact Username Passthrough enhancement for non-IMS networks. See Information About Contact Username Passthrough, page 22-6.

## Adding an Expires-Header to a Register-Message

Some registrars or endpoints might fail to understand the Expires parameters configured in the contact URI. To overcome this issue, you can configure the SBC to add an Expires header to the register messages.

In SIP, the expiry time of a registration is specified by registering the endpoints using the following:

- Expires header
- The Expires parameter on the registered contact URI.
- The registrar can choose an expiration period, if no expirations period is specified.

## Configuring SBC to Add an Expires-Header

To configure SBC to add an Expires header to the register messages, complete the following steps:

**SUMMARY STEPS**

1. configure
2. sbc service-name
3. sbe
4. adjacency sip adjacency-name
5. expires-header
6. softswitch-shield
7. exit
8. end
9. show sbc sbc-name sbe adjacencies adjacency-name Detail
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc service-name</td>
<td>Enters the mode of an SBC service. Use the <code>service-name</code> argument to define the name of the service.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# sbc mysbc</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Enters the mode of an SBE entity within an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> adjacency sip adjacency-name</td>
<td>Enters the mode of an SBE SIP adjacency. Use the <code>adjacency-name</code> argument to define the name of the service.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe)# adjacency sip sipGW</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> expires-header options</td>
<td>Adds the Expires parameter in a SIP contact header. Use the <code>options</code> argument to specify one of the following strings for adding expires to the header:</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-adj-sip)# expires-header add-not-present</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> softswitch-shield</td>
<td>Enables softswitch shielding on the SIP.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-adj-sip)# softswitch-shield</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> exit</td>
<td>Exits the adj-sip-ping mode, and moves to adj-sip mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-adj-sip-ping)# exit</td>
<td></td>
</tr>
</tbody>
</table>
Support for Supported Path Under REGISTER Request

Starting with Cisco IOS XE Release 2.5, Cisco Unified Border Element (SP Edition) supports the use of the Path extension header field in the Supported field of a REGISTER Request. The Path field provides a way to accumulate and send a list of proxies between a SIP user agent and a registrar. For information on the Path field, see RFC 3327 Session Initiation Protocol (SIP) Extension Header Field for Registering Non-Adjacent Contacts.

Information About Contact Username Passthrough

The Contact Username Passthrough feature enables interoperability with softswitches that require the contact username portion of the Contact URI in SIP REGISTER requests to pass through unchanged. In certain situations in non-IMS networks, a softswitch may be unable to operate with Cisco Unified Border Element (SP Edition) rewriting or hashing the contact username portion of the Contact URI in SIP REGISTER requests. In these cases, subscribers may not be able to register through the SBC unless you configure the SBC to pass through the contact username.

In a typical SIP registration process, the default behavior is that Cisco Unified Border Element (SP Edition) rewrites the URIs in the Contact headers of REGISTER requests sent by subscribers for the following reasons:

- To remain on the signaling path for requests sent to this subscriber.
- To disambiguate subscribers that register from different devices with the same private username by replacing the username part of the Contact URI with a unique string.

For example, the SBC receives two REGISTER requests, with the following contact URIs using the same username, “bob”:

bob@1.1.1.1
bob@2.2.2.2

In each case, the REGISTER contains a Contact URI with the SBC’s address. The SBC replaces or rewrites the username “bob” in each URI with a unique string to disambiguate them.

In Cisco IOS XE Release 2.5 and later, you can choose to configure the SBC to pass through, not rewrite, the contact username on SIP REGISTER requests by ensuring that each contact username associated with a given subscriber uses a different port number. By using unique ports for each contact sent to the registrar, the SBC can uniquely correlate to the registered endpoints without requiring a unique username. This can be configured for each adjacency facing the registrar.

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 8 end</td>
<td>Exits the SBE mode and returns to the privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe)# end</td>
<td></td>
</tr>
</tbody>
</table>

| Step 9 show sbc sbc-name sbe adjacencies adjacency-name detail | Lists the configured Expires headers for the specified adjacency. |
| Example: |         |
| Router# show sbc mysbc sbe adjacencies sipGW detail |         |
The following is an example of contact username passthrough where the username “bob” is passed through unchanged and the hostport is rewritten with the address of the SBC and a unique port number:
sip:bob@1.1.1.1 ----> sip:bob@192.168.101.1:5060
See the Contact Username Passthrough Examples section on page 22-38 for more configuration examples.

Note
This feature has no effect in IMS deployments where the SBC does not rewrite contact usernames.

Configuring Contact Username Passthrough

You can use the registration contact username passthrough and the signaling-port commands in the (config-sbc-sbe-adj-sip) configuration mode to configure the Contact Username Passthrough feature.

The registration contact username command with the passthrough option allows you to specify that the contact username in the SIP REGISTER request should be passed through unchanged when rewriting contacts. This option should be enabled on the registrar-facing adjacency. The passthrough option disambiguates subscribers that register from different devices with the same private username by using a unique local port number when multiple contact URIs are registered for the same public ID. The range of valid signaling ports are configured with the signaling-port command on a registrar-facing adjacency.

If you do not specify the max-port-num option in the signaling-port command on this adjacency, then the SBC is not able to disambiguate subscribers that register from different devices with the same username.

The default is the rewrite option which allows the username to be changed when rewriting contacts.

Note
If the contact username is longer than 32 characters, then it is not passed through and is replaced with a hash as is the case when the default rewrite option is chosen.

The following example configures the SBC to specify that the contact username is passed through unchanged.

SUMMARY STEPS

1. configure terminal
2. sbc sbc-name
3. sbe
4. adjacency sip adjacency-name
5. no attach
6. registration contact username {passthrough | rewrite}
7. signaling-port port-num [max-port-num]
8. exit
9. end
10. show sbc sbe adjacencies
## Information About Contact Username Passthrough

### Detailed Steps

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>configure terminal</td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>sbc sbc-name</td>
<td>Creates the SBC service on the SBC and enters into SBC configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config)# sbc mySbc</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>sbe</td>
<td>Enters the mode of the signaling border element (SBE) function of the SBC.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>adjacency sip adjacency-name</td>
<td>Configures the adjacency (facing the registrar), and enters into adjacency sip configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe)# adjacency sip adj1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>no attach</td>
<td>(Optional) Use this command to detach an existing adjacency so it is not active before modifying it.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-adj-sip)# no attach</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>registration contact username (passthrough</td>
<td>rewrite)</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-adj-sip)# registration contact username passthrough</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>signaling-port port-num [max-port-num]</td>
<td>Configures range of valid signaling ports on a registrar-facing adjacency to allow the SBC to disambiguate subscribers that register from different devices with the same username. max-port-num is the range from 1 through 65535. If both port-num and max-port-num are specified, then the port-num indicates the lower boundary of the range and max-port-num indicates the upper boundary of the range. If no max-port-num is specified, then the adjacency listens only on the single port-num. Max-port-num only needs to be set if a range of local listen ports is required for this adjacency.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-adj-sip)# signaling-port 5060 5062</td>
<td></td>
</tr>
</tbody>
</table>
Chapter 22 Cisco Unified Border Element (SP Edition) Registration Features

Information About Alternative Contact Rewriting

In a User-to-Network Interface (UNI) deployment scenario, the endpoint registers to registrar, softswitch, or proxy through the SBC. The SBC rewrites the received contact header to use its own signaling address and keeps itself in the call signaling flow. The SBC maintains the mapping between the received and forwarded contact information. This ensures that the contacts received from the multiple devices are unique, and provides anonymity to the subscribers.

Prior to Cisco IOS XE Release 3.3S, the SBC would rewrite the contact by replacing the entire contact with an alphanumeric string generated by hashing the received contact information. However, registrars can determine that the multiple registrations are for the same Address of Record (AOR) by comparing an initial section of the user-info in the contact header. For example, they determine that the two contacts for 02083661177-abc@sbc.com and 02083661177-xyz@sbc.com are for the same endpoint.

From Cisco IOS XE Release 3.3S, the SBC rewrites the contact header in the following two methods:

1. Hashed value of hexadecimal characters—<DN> + "." + <hashed_value>, the <hashed_value> is a randomly generated value and is unique for a specific endpoint, so that the softswitch can identify those endpoints and initiate forking. Forking is an multiple calls attempt to the endpoints for a single AoR.
2. Username of rewritten contact Uniform Resource Identifier (URI) only includes numeric hashed value.

Delegate Registration

When a delegate subscriber is configured on a preset-access adjacency, the contact header sent to the registrar is rewritten similar to the contact header received on a REGISTER message. Therefore, the Alternative Contact Rewriting feature applies to a delegate registration also. The original contact provided to the user on exit and used to generate the rewritten contact is the contact that is configured using the **sip-contact contact uri** command under the SBE Subscriber Entry mode.

Restrictions on Alternative Contact Rewriting

The feature has the following restriction:

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 8  <strong>exit</strong></td>
<td>Exits adjacency sip configuration mode and enters into SBE configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-adj-sip)# exit</td>
<td></td>
</tr>
<tr>
<td>Step 9  <strong>end</strong></td>
<td>Exits SBE configuration mode and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe)# end</td>
<td></td>
</tr>
<tr>
<td>Step 10 <strong>show sbc sbe adjacencies</strong></td>
<td>Displays SBC adjacencies.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# show sbc sbe adjacencies</td>
<td></td>
</tr>
</tbody>
</table>
If there is no username in a contact URI, 32 digits hashed username is used. However, if the original username is 24 bytes or more in length, the username is rewritten in the `<23 digit numeric hash>-<8 digit numeric hash>` format.

### Configuring Alternative Contact Rewriting

This task explains how to configure the Alternative Contact Rewriting feature.

#### SUMMARY STEPS

1. configure
2. sbc sbc-name
3. sbe
4. adjacency {sip | h323} adjacency-name
5. registration contact username rewrite [numeric | userid-and-numeric]
6. end
7. show sbc sbc-name sbe adjacencies adjacency-name detail

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>configure terminal</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;Router# configure terminal</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><strong>sbc sbc-name</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;Router(config)# sbc mySbc</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><strong>sbe</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;Router(config-sbc)# sbe</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>**adjacency {sip</td>
</tr>
</tbody>
</table>
# Information About Alternative Contact Rewriting

## Step 5

**Command or Action**

```
registration contact username rewrite [numeric | userid-and-numeric]]
```

**Example:**

```
Router(config-sbc-sbe-adj-sip)# registration contact username rewrite userid-and-numeric
```

**Purpose**

Configures the contact username in a SIP REGISTER request so that it can be modified.

- **rewrite**—Allows the contact username in a SIP REGISTER request to be changed or rewritten.
- **numeric**—Rewrites the contact username in a SIP REGISTER request as an originating hashed numeric value.
- **userid-and-numeric**—Rewriting the contact username in a SIP REGISTER request as an originating userid plus hashed numeric value.

## Step 6

**Command or Action**

```
end
```

**Example:**

```
Router(config-sbc-sbe)# end
```

**Purpose**

Exits adjacency SIP configuration mode and returns to Exec mode.

## Step 7

**Command or Action**

```
show sbc sbe-name sbe adjacencies adjacency-name detail
```

**Example:**

```
Router# show sbc sbe mySBC sbe adjacencies pe42
```

**Purpose**

Displays the detailed field output for the specified SIP adjacency.

The following example show the output of the `show sbc sbe adjacencies detail` command:

```
Router# show sbc pe41 sbe adjacencies pe42 detail
SBC Service "pe41"
Adjacency pe42 (SIP)
  Status: Attached
  Signaling address: 88.41.41.41:5060
  IPsec server port: 0
  Signaling-peer: 88.42.42.42:5060 (Default)
  Signaling-peer status: Not Tested
  Signaling-peer priority: 2147483647
  Signaling-peer switch: always
  Peer status: Not Tested
  Current peer index: 0
  Force next hop: No
  Force next hop select: Out-of-dialog
  Admin Domain: None
  Account:
    Group: None
  
  Rewrite REGISTER: Off
  Register contact username: Rewrite as userid and digits
  Target address: None
  NAT Status: Auto Detect
  Reg-min-expiry: 3000 seconds
  Local Jitter Ratio: 0/1000
  
```
Information About SIP Fast Registration

SIP Fast Registration performs the following functions:

- Prompts endpoints to register frequently with Cisco Unified Border Element (SP Edition), causing NAT/Firewall pinholes to remain open.
- Protects internal network elements from the large number of REGISTER messages arising from endpoint registration.
- Allows Cisco Unified Border Element (SP Edition) to do minimal processing of REGISTER messages, which improves performance. This is particularly important when dealing with significant oversubscription—where there typically may be 10 times more subscribers than active calls.

When Cisco Unified Border Element (SP Edition) faces the customer premise side and the customer on the network edge deploys Network Address Translation (NAT) and firewalls, SIP INVITEs to the NAT endpoint are unable to penetrate the NAT to establish calls. To overcome the problem, the endpoints transmit SIP REGISTER requests at a high enough frequency to keep the NAT pinhole alive. The SIP Fast Registration feature off-loads the processing from the registrar for a large number of endpoints and generates SIP REGISTER replies from the data plane (forwarding services provided by Cisco QuantumFlow Processor (QFP)). This also limits the impact on the router’s CPU load. The QFP processes the expected SIP re-register messages by short-cutting the REGISTER messages and quickly turning them around.

Typically the registrar responds to the first SIP REGISTER message asking the end-point to send its next SIP REGISTER message within 3600 seconds (configurable). With the SIP Fast Registration feature, Cisco Unified Border Element (SP Edition) intercepts this Reply and informs the end-point to REGISTER every 30 seconds to keep the NAT open. The Route Processor (RP) programs a SIP Fast Registration (SFX) entry with a fast expiry times parameter in seconds. When the fast expiry times parameter expires, the QFP punts the SIP REGISTER message to the RP to update the state before forwarding it to the registrar.

Fast registration is configured per SIP adjacency, on the endpoint-facing adjacency, that is, the adjacency which receives the incoming REGISTER request. After an endpoint has registered using fast registration through an adjacency, all subsequent registration requests from the same endpoint are responded to by Cisco Unified Border Element (SP Edition), without notifying the softswitch, until the registration interval has almost expired.

The following shows a fast registration configuration example:

```
...  
Reg-min-expiry: 300 seconds
Fast-register: Enabled
Fast-register-interval: 60 seconds
Register aggregate: Disabled
...  
```

In the example above, Cisco Unified Border Element (SP Edition) performs fast registrations at 60 second intervals, and will send registrations towards the registrar/softswitch at intervals of 480 seconds. This is calculated by a hard-coded algorithm of (3 x fast-register-interval) + (reg-min-expiry), plus taking into account the expiry time in the inbound registration from the endpoint.

In the above example, Softswitch Shielding is not enabled. Thus if incoming expiry time from the endpoint is 400 seconds, which is less than 480 seconds, then the incoming registration interval to the registrar/softswitch is calculated as 480 seconds. However, if the incoming expiry time from the endpoint is 600 seconds, which is larger than 480 seconds, then the incoming registration interval to the registrar/softswitch is calculated as 600 seconds.
On the other hand, when Softswitch Shielding is enabled, then the Softswitch Shielding timer takes precedence and is always used as the incoming registration interval to the registrar/softswitch. See the Information About SoftSwitch Shielding section on page 22-16.

When fast registration is enabled, the incoming registration time should not be less than the fast-register-interval. Otherwise, the SBC will reject the registration with error message 423 (Interval Too Brief). The SBC compares incoming registration time with the interval set in fast-register-interval, instead of the interval in reg-min-expiry. If fast registration is disabled, then the incoming registration time should not be less than the reg-min-expiry time. Otherwise, the SBC will reject the registration with response code 423 (Interval Too Brief).

For information on commands, such as fast-register-interval and reg-min-expiry, see the Cisco Unified Border Element (SP Edition) Command Reference: Unified Model.

SIP Fast Registration is not enabled by default. You must configure it with the inherit profile preset-access command, either as a global configuration or per adjacency.

Once SIP Fast Registration is configured, fast-pathing is on by default on an adjacency. You can then disable Fast Registration using the fast-register disable command.

REGISTER messages are rejected by Cisco Unified Border Element (SP Edition) with a 423 Interval Too Brief response code under the following conditions:

- If Fast Registration is enabled and the Expires header in the REGISTER message is less than the inbound adjacency’s “fast-register-interval.”
  
  The fast-register-interval command controls the recommended rate at which endpoints send REGISTER requests. The lower this value, the more frequently endpoints re-register, thus keeping a NAT/firewall pinhole open. Therefore, we recommend configuring this to a slightly lower value than the pinhole timeout, if that is known.

- If Fast Registration is not enabled and the Expires header in the REGISTER messages is less than the inbound adjacency’s “reg-min-expiry.”
  
  The reg-min-expiry command controls the rate at which REGISTER requests are sent from the SBC to the SIP registrar. The lower this value, the greater the potential register load on the softswitch. If fast-pathing is not enabled for an adjacency, SBC rejects any REGISTER requests with a shorter expiry interval than the reg-min-expiry command.
Figure 22-1 illustrates where network elements are located in a network configured with Fast Registration, SoftSwitch Shielding, and Aggregate Registration.

**Figure 22-1 Voice Network Elements in a Fast Registration, SoftSwitch Shielding, and Aggregate Registration Network**

Restrictions for SIP Fast Registration

The restrictions for SIP Fast Registration are the following:

- Only UDP is supported.
- REGISTERs with a zero expiry time ("Unregisters") are always forwarded to the registrar and not fast-patched, if the SBC matches them to a known registration.
- Minimal parsing of REGISTER requests is performed before a decision is taken to send a fast-path response; this minimizes the load on the SBC. A REGISTER request is only fast-patched if its expiry interval is not zero and it comes from the same IP address and port as a known subscription.
- Endpoints that send their requests from ephemeral (short-lived) ports do not have their registration requests fast-patched.
- The fast-register-interval cannot be higher than the reg-min-expiry (minimum expiry value). If the minimum expiry value is less than twice the fast-register-interval, fast-pathing is not performed.
Configuring SIP Fast Registration

This task configures a basic SIP Fast Registration on an adjacency.

SUMMARY STEPS

1. configure
2. sbc sbc-name
3. sbe
4. adjacency {sip | h323} adjacency-name
5. inherit profile {preset-access | preset-core | preset-ibcf-ext-untrusted | preset-ibcf-external | preset-ibcf-internal | preset-p-cscf-access | preset-p-cscf-core | preset-peering | preset-standard-non-ims}
6. exit
7. end
8. show platform hardware qfp active feature sbc sfx

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>configure terminal</strong> Enables global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router# configure terminal</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><strong>sbc sbc-name</strong> Creates the SBC service on the SBC and enters into SBC configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config)# sbc mySbc</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><strong>sbe</strong> Enters the mode of the signaling border element (SBE) function of the SBC.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc)# sbe</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>**adjacency {sip</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe)# adjacency sip adj1</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>H.323 adjacencies are not supported in Cisco IOS XE Release 2.4 and earlier.</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>**inherit profile {preset-access</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-adj-sip)# inherit profile preset-access</td>
</tr>
</tbody>
</table>
Information About SoftSwitch Shielding

Cisco Unified Border Element (SP Edition) supports the SoftSwitch Shielding feature that allows a lower SIP registration rate on the links to registrars (typically softswitches) than on the links to endpoints. In a network where endpoints frequently refresh their SIP registrations to a softswitch, allowing a lower registration rate shields the softswitch from an undesirably high rate of re-registrations while ensuring the softswitch still has accurate knowledge of registered endpoints.

For example, if endpoints are sending REGISTER messages to inform the proxy server of the callee address location every 15 minutes, Cisco Unified Border Element (SP Edition) can be configured to only forward REGISTER messages to the softswitch every 12 hours, unless there is a change to the contact being registered. Using SoftSwitch Shielding reduces the load on the softswitch and the network. On the other hand, if an endpoint stops sending REGISTER messages, Cisco Unified Border Element (SP Edition) detects the change within the endpoint’s expiry interval and removes the subscriber state, thus preventing calls to or from this endpoint.

The SoftSwitch Shielding feature gives Cisco Unified Border Element (SP Edition) the capability to shield the softswitch from a large portion of the registration processing. You are also able to simultaneously configure the SIP Fast Registration feature and the SoftSwitch Shielding feature. In addition, if the REGISTER message contains an Authorization header, Cisco Unified Border Element (SP Edition) forwards the REGISTER message to the softswitch registrar.

Use the `registration outgoing timer` command to enable SoftSwitch Shielding and set the time interval during which the SBC forwards REGISTER messages to the softswitch before timing out.

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 6 <code>exit</code></td>
<td>Exits adjacency sip configuration mode and enters into SBE configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-adj-sip)# exit</td>
</tr>
<tr>
<td>Step 7 <code>end</code></td>
<td>Exits SBE configuration mode and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe)# end</td>
</tr>
<tr>
<td>Step 8 <code>show platform hardware qfp active feature sbc sfx</code></td>
<td>Displays the QFP SIP Fast-Register (SFX) counters.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router# show platform hardware qfp active feature sbc sfx global</td>
</tr>
</tbody>
</table>
Figure 22-2 illustrates a SoftSwitch Shielding call flow.

**Figure 22-2  SoftSwitch Shielding Call Flow**

![SoftSwitch Shielding Call Flow Diagram](image)

### Configuring SoftSwitch Shielding

This task configures SoftSwitch Shielding on an adjacency.

**SUMMARY STEPS**

1. `configure`
2. `sbc sbc-name`
3. `sbe`
4. `adjacency {sip | h323} adjacency-name`
5. `registration outgoing timer sec`
6. `registration rewrite-register`
7. `inherit profile {preset-access | preset-core | preset-ibcf-ext-untrusted | preset-ibcf-external | preset-ibcf-internal | preset-p-cscf-access | preset-p-cscf-core | preset-peering | preset-standard-non-ims}`
**8. exit**
9. `end`
10. `show sbc sbc-name sbe adjacencies adjacency-name detail`

## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc <code>sbc-name</code></td>
<td>Creates the SBC service on the SBC and enters into SBC configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# sbc mySbc</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Enters the mode of the signaling border element (SBE) function of the SBC.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> adjacency `{sip</td>
<td>h323}<code> </code>adjacency-name`</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe)# adjacency sip SoftSwitch</td>
<td><strong>Note</strong> H.323 adjacencies are not supported in Cisco IOS XE Release 2.4 and earlier.</td>
</tr>
<tr>
<td><strong>Step 5</strong> registration outgoing timer <code>sec</code></td>
<td>Enables SoftSwitch Shielding and sets the registration timeout timer for the time interval during which the SBC forwards outgoing REGISTER messages to the softswitch before timing out.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-adj-sip)# registration outgoing timer 36000</td>
<td><code>sec</code>—value is 1 to 2147483647. The default of zero disables SoftSwitch Shielding.</td>
</tr>
<tr>
<td><strong>Step 6</strong> registration rewrite-register</td>
<td>Configures the SIP register request rewriting on an adjacency.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-adj-sip)# registration rewrite-register</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> inherit profile `{preset-access</td>
<td>preset-core</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-adj-sip)# inherit profile preset-core</td>
<td>An adjacency facing the registrar typically has a preset-core profile.</td>
</tr>
<tr>
<td></td>
<td>The default is preset-core.</td>
</tr>
</tbody>
</table>
### Information About Registration Monitoring

Cisco Unified Border Element (SP Edition) supports creating event subscriptions for changes of registration state. Event subscriptions are generally network-initiated de-registrations. This is a requirement of the IP Multimedia Subsystem (IMS) specifications for Proxy-Call Session Control Function (P-CSCF), a SIP proxy server. For more information, refer to 3rd Generation Partnership Project (3GPP) TS 24.229 v7.5.1.

This support is configured on a per-adjacency basis through the registration monitor field of the Adjacency Table. If this field is set, then Cisco Unified Border Element (SP Edition) creates an event subscription with the registrar for each registered subscriber situated on the adjacency.

The registrar uses the event subscription to provide active indications of changes to the state of the registration. Based on these indications, Cisco Unified Border Element (SP Edition) adds, removes, or updates subscriber state, as appropriate. For more information on event subscriptions, refer to RFC 3680.

Cisco Unified Border Element (SP Edition) does not clean up fast register configuration in the event of a network-initiated de-registration. In this case, the user equipment (UE) is not able to re-register with the registrar until the fast register time period expires.

Cisco Unified Border Element (SP Edition) sets the duration of the monitoring subscription to be the maximum expired interval of the subscriber’s contacts plus a configurable constant. The default monitoring subscription duration is 32 seconds.

Cisco Unified Border Element (SP Edition) only re-subscribes to the Serving-Call Session Control Function’s (S-CSCF) monitoring state when the UE sends a re-register through the SBC. Cisco Unified Border Element (SP Edition) does not follow the 3GPP model of refreshing subscriptions 600 seconds before they expire. The SBC implementation reduces the load on the P-CSCF and S-CSCF, both SIP servers, while ensuring that the subscription lifetime exceeds the registration lifetime, which ensures that network-initiated de-registrations are always detected.

Use the `registration monitor` command to enable monitoring of event subscriptions for registration state changes.

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 8**

`exit`

**Example:**

```
Router(config-sbc-sbe-adj-sip)# exit
```

Exits adjacency sip configuration mode and enters into SBE configuration mode.

| **Step 9**

`end`

**Example:**

```
Router(config-sbc-sbe)# end
```

Exits SBE configuration mode and returns to Exec mode.

| **Step 10**

`show sbc sbc-name sbe adjacencies adjacency-name detail`

**Example:**

```
Router# show sbc sbc mySBC sbe adjacencies
SoftSwitch detail
```

Displays all the detailed field output for the specified SIP adjacency, including the “Register Out Timer:” field that shows the time interval in seconds when the SBC forwards the next REGISTER messages to the softswitch.
### Configuring Registration Monitoring

This task configures how to monitor event subscriptions as a result of registration state changes.

#### SUMMARY STEPS

1. `configure`
2. `sbc sbc-name`
3. `sbe`
4. `adjacency {sip | h323} adjacency-name`
5. `registration monitor`
6. `exit`
7. `end`
8. `show sbc sbc-name sbe adjacencies adjacency-name detail`

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td><code>configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Creates the SBC service on the SBC and enters into SBC configuration mode.</td>
</tr>
<tr>
<td><code>sbc sbc-name</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# sbc mySbc</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Enters the mode of the signaling border element (SBE) function of the SBC.</td>
</tr>
<tr>
<td><code>sbe</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Configures the adjacency facing the registrar, and enters into adjacency sip configuration mode.</td>
</tr>
<tr>
<td>`adjacency {sip</td>
<td>h323} adjacency-name`</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe)# adjacency sip Cary-IP-PBX</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Enables monitoring of event subscriptions as a result of registration state changes.</td>
</tr>
<tr>
<td><code>registration monitor</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-adj-sip)# registration monitor</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>Exits adjacency sip configuration mode and enters into SBE configuration mode.</td>
</tr>
<tr>
<td><code>exit</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-adj-sip)# exit</td>
<td></td>
</tr>
</tbody>
</table>
Information About Per Subscriber Delete

The Per Subscriber Delete feature provides a mechanism for manually deleting individual subscribers, and associated registered contacts or other subscriber state if any, from the database. This feature works on both manually created subscriber entries and dynamically created entries during the standard registration process. It does not cause the SBC to signal either the subscriber or the registrar; it only removes the internal state from the SBC associated with that subscriber.

Use the `clear sbc sbe sip subscriber aor` command to clear the stuck registrations on Cisco ASR 1000 Series Routers.

Configuring Per Subscriber Delete

This section shows how to clear stuck registrations.

### SUMMARY STEPS

1. `show sbc sbe-name sbe sip subscribers`
2. `clear sbc sbe-name sbe sip subscriber aor address-of-record`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1**

- `show sbc sbe-name sbe sip subscribers`
  
  **Example:**
  
  Router# show sbc asr sbe sip subscribers

  Displays details of all the SIP endpoints that have registered with the SBC. Information about the Address of Record (AOR) for each subscriber is also displayed.

| **Step 2**

- `clear sbc sbe-name sbe sip subscriber aor address-of-record`
  
  **Example:**
  
  Router# clear sbc asr sbe sip subscriber aor sip:alice@open-ims.test

  Clears the stuck registrations on Cisco ASR 1000 Series Routers.
Information About Aggregate Registration

A registrar is typically a registration server in a SIP network, but outside of the Cisco Unified Border Element (SP Edition) device. The registrar accepts and processes registration requests that register one or more IP addresses to a specific URI, usually a “sip:” address. Because SIP endpoints need to know each others IP address, the registrar acts as a location service. More than one user agent can register at the same IP address. When a call is placed to that IP address, all the registered user agents will ring.

Cisco Unified Border Element (SP Edition) supports Aggregate Registration where a single registration is implemented that causes the registrar to implicitly register multiple IP addresses. The SBC performs aggregate registration for endpoints connected to it. Thus Cisco Unified Border Element (SP Edition) can support devices that implicitly registers multiple endpoints through it. This way of registering all endpoints connected to it in a single registration can be compared to what is commonly done by an IP-PBX device.

The Aggregate Registration feature allows single registration on behalf of multiple endpoints and implicit registration of single endpoints behind the Internet Protocol Private Branch eXchange (IP-PBX).

Aggregate registration is configured on a per-adjacency basis and is configured under an adjacency. All end user clients attached to the adjacency can perform aggregate registration.

When an adjacency has aggregate registration support enabled, the SBC behaves as follows:

- On receiving a REGISTER message, the SBC caches the top Via header and stores it with the normal registration details.
- On receiving an INVITE or out-of-dialog request on the adjacency, Cisco Unified Border Element (SP Edition) attempts to look up the registration using the top Via header, not the Contact and From headers. This ensures that the SBC routes the call to the registrar correctly.
- On receiving an INVITE or out-of-dialog request to the adjacency, Cisco Unified Border Element (SP Edition) overwrites the Request URI as follows:
  - The username is overwritten with the username in the P-Called-Party-Id header if present, or the To header if not.
  - The hostname is overwritten with the hostname that was present on the Contact header that the PBX registered.

Use the registration aggregate command to enable Aggregate Registration support from an adjacency.

Configuring Aggregate Registration

This task configures Aggregate Registration on an adjacency.

SUMMARY STEPS

1. configure
2. sbc sbc-name
3. sbe
4. adjacency [sip | h323] adjacency-name
5. registration rewrite-register
**Information About Aggregate Registration**

6. inherit profile {preset-access | preset-core | preset-ibcf-ext-untrusted | preset-ibcf-external | preset-ibcf-internal | preset-p-cscf-access | preset-p-cscf-core | preset-peering | preset-standard-non-ims}

7. registration aggregate

8. header-name [contact [add [tls-param]] | from{passthrough} | to{passthrough}]

9. request-line request-uri rewrite

10. exit

11. end

12. show sbc sbc-name sbe adjacencies adjacency-name detail

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc sbc-name</td>
<td>Creates the SBC service on the SBC and enters into SBC configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config)# sbc mySbc</td>
</tr>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Enters the mode of the signaling border element (SBE) function of the SBC.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc)# sbe</td>
</tr>
<tr>
<td><strong>Step 4</strong> adjacency {sip</td>
<td>h323} adjacency-name</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe)# adjacency sip Cary-IP-PBX</td>
</tr>
<tr>
<td><strong>Step 5</strong> registration rewrite-register</td>
<td>Configures the SIP register request rewriting on an adjacency.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-adj-sip)# registration rewrite-register</td>
</tr>
<tr>
<td><strong>Step 6</strong> inherit profile {preset-access</td>
<td>preset-core</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-adj-sip)# inherit profile preset-access</td>
</tr>
</tbody>
</table>

**Note** H.323 adjacencies are not supported in Cisco IOS XE Release 2.4 and earlier.
Information About Provisioned Delegate Registration

In a SIP network, some third-party client or end user devices are unable to register themselves with a registrar. A registrar is a server that resides outside the SBC device. These end users or clients are generally the applications running on systems used by people. The application may be a softphone application running on your PC or a messaging device in your IP phone. For example, the softphone application generates a request when you try to call another person over the network and sends the request to a server. The generated request is a register message or registration.

End users register their locations or addresses to a registrar server. By registering or sending a special message to a registrar server, the registrar server maintains updated locations of end users in a SIP network. The client then sends the request to a proxy server because when the request is generated, the address of the recipient or callee is not known. The registration is sent to inform a proxy server of the location of the callee address.
Cisco Unified Border Element (SP Edition) can be configured to register any client devices that cannot register themselves. Using the Provisioned Delegate Registration feature, you can set up delegate registration for individual client devices. The client device can make and receive calls as though it had registered normally. Additionally, you can specify individual parameters for each client device, such as registering the client device to a specified registrar server.

Provisioned Delegate Registration is done by provisioning Cisco Unified Border Element (SP Edition) with enough information about a client device so that it can originate a registration for the device itself. Cisco Unified Border Element (SP Edition) can perform end user registration for up to several hundred to a thousand end user clients or delegate clients.

Provisioned Delegate Registration supports the following functionalities:

- Cisco Unified Border Element (SP Edition) can be configured to register with a SIP registrar on behalf of another SIP network entity – the delegate client.
- Users can configure the following information per delegate client:
  - The registrar where the end user client is registered.
  - The registrar facing adjacency if the user wants to bypass normal routing.
  - The adjacency facing the delegate client.
  - The expiration time of the registration.
  - The refresh time of the registration.
  - The contact Uniform Resource Identifier (URI) information of the delegate client. Depending on the adjacency configuration, the URI may be rewritten on messages going to the registrar to force calls to the delegate client to be routed through the SBC.
  - The next hop to the delegate client (for calls coming from the registrar).
  - The address of record (AoR) of the delegate client.
  - The number of attempts and frequency of registration retries when a failure is received.
- Calls from the delegate client can be forwarded to one of the following (in order of preference):
  - the entity identified in the Service-Route header if present on the registration response, or
  - the registrar.
- Calls to the delegate client must have the Request URI rewritten to indicate the delegate client, and forwarded out of the adjacency facing it.

Restrictions

The following is a restriction of the Provisioned Delegate Registration feature:
- The delegate registration configuration is limited to no more than 1000 subscribers with each subscriber having no more than 5 contacts.

Provisioned Delegate Registration Call Flow Description

When an end user client is brought up, Cisco Unified Border Element (SP Edition) builds a REGISTER message on behalf of the client, and sends it to the specified registrar. The REGISTER message contains all the contact Uniform Resource Identifier (URI) information that have been configured on the end user client.
Information About Provisioned Delegate Registration

If the registrar responds positively to the REGISTER message, Cisco Unified Border Element (SP Edition) stores this fact. Calls to and from the end user client is treated by the SBC exactly as if the client has registered itself.

If the registrar responds negatively to the REGISTER message. For example, if Cisco Unified Border Element (SP Edition) receives error response 423—Interval too brief, the SBC retries building a REGISTER message after the configured retry interval. Cisco Unified Border Element (SP Edition) repeats this process the configured number of times. If the end user client has still failed to be registered, a log is made and the subscriber operating status is changed to OPER_ACT_FAILED.

Before the registration time of the end user client expires, Cisco Unified Border Element (SP Edition) performs the registration processing again to refresh the callee/recipient address location through registration.

See Provisional Delegate Registration Commands section on page 22-26 for more information on configuration steps and commands.

Configuring Delegate Registration Profile

Cisco Unified Border Element (SP Edition) requires a configured subscriber for each end user client upon whose behalf the SBC performs registration. The user may configure retry counts, retry intervals, duration, and the refresh buffer time for each configured subscriber, also called “delegated subscriber.” Several subscribers may all share the same nondefault values for the fields in the Delegate Registrations (amb_mw_sudb_subscriber) table. Instead of requiring configuration for each subscriber separately, Cisco Unified Border Element (SP Edition) allows the user to configure a subscriber profile that can be applied to one or more subscribers.

Use the delegate-profile profile-name command to configure a profile for a delegate registration subscriber.

Provisional Delegate Registration Commands

When the AdminStatus of the Delegate Registrations table is set to AdminStatusUp, Cisco Unified Border Element (SP Edition) attempts to register with the registrar. If the registration is successful, the delegate/client device is treated the same as all other subscribers. Cisco Unified Border Element (SP Edition) registers the device for the length of time specified. Cisco Unified Border Element (SP Edition) renews the registration before it expires, with the specified configurable buffer time.

If a registration (or re-registration) fails, Cisco Unified Border Element (SP Edition) retries registration after the configured delay time. Cisco Unified Border Element (SP Edition) retries the specified number of times. If registration still fails, Cisco Unified Border Element (SP Edition) logs the failure and sets the subscriber operating status to failed.

The following commands are used to configure Provisional Delegate Registration:

• Use the delegate-profile command to configure a delegate client registration profile that can be applied to a delegate subscriber. After a delegate profile is configured, profile parameters that specify duration, retry-count, retry-interval, and refresh-buffer may optionally be configured.

• Use the subscriber aor command to define the address of record for the subscriber and define the unique subscriber for whom you want to configure delegate registration. The subscriber must have one or more SIP contacts/URIs associated with it.
• Use the `sip-contact uri` command to configure a SIP contact URI for a subscriber. The contact information is used to provision Cisco Unified Border Element (SP Edition) with client device information, so the SBC can register the device. For every delegate registration configured with the `delegate-registration hostname` command, one or more SIP contacts/URIs must be configured in the SIP Contacts table (amb_mw_subd_local_id).

• Use the `delegate-registration hostname` command to configure a delegate registration for a delegate client.

• Use the `profile` command to apply a delegate registration profile to a delegate registration subscriber.

• Use the `show sbc sbc-name sbe sip subscribers` command to display subscribers for whom Provisioned Delegate Registration has been provisioned.

• Use the `show sbc sbc-name sbe sip delegate-profiles` command to display subscriber profiles for whom Provisioned Delegate Registration has been configured.

### Configuring Provisional Delegate Registration

This task configures in order: a profile for a delegate registration subscriber; delegate registration for a specified subscriber associated with an individual client device; delegate registration for a specified client/delegate device; and displays subscribers and subscriber profiles for whom delegate registration have been configured.

#### SUMMARY STEPS

1. configure
2. `sbc sbc-name`
3. `sbe`
4. `delegate-profile profile name`
5. `duration dur time in secs`
6. `retry-count #times to retry`
7. `retry-interval retry time in secs`
8. `refresh-buffer timeout in secs`
9. `exit`
10. `subscriber aor`
11. `sip-contact uri`
12. `adjacency adjacency name`
13. `exit`
14. `delegate registration hostname`
15. `adjacency adjacency name`
16. `profile my-profile`
17. `activate`
18. `end`
19. `show sbc sbc-name sbe sip subscribers delegate`
20. `show sbc sbc-name sbe sip delegate-profiles`
## Information About Provisioned Delegate Registration

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td></td>
</tr>
<tr>
<td><strong>configure</strong></td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>Router# configure</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td></td>
</tr>
<tr>
<td><code>sbc sbc-name</code></td>
<td>Creates the SBC service on the SBC and enters into SBC configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>Router(config)# sbc mySbc</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td></td>
</tr>
<tr>
<td><code>sbe</code></td>
<td>Enters the mode of the signaling border element (SBE) function of the SBC.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>Router(config-sbc)# sbe</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td></td>
</tr>
<tr>
<td><code>delegate-profile profile name</code></td>
<td>Configures a delegate/client registration profile that can be applied to a delegate registration subscriber. Enters into subscriber delegate profile configuration mode where profile parameters can be configured. The profile name is a string of 24 characters maximum length.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>Router(config-sbc-sbe)# delegate-profile my profile</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td></td>
</tr>
<tr>
<td><code>duration dur time in secs</code></td>
<td>Configures the expiration time when the delegate client is due to expire, that is, the length of time in seconds during which the SBC tries to perform delegate registration before stopping. The default duration time is 1800 seconds. The range is 1 to 2,147,483 seconds.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>Router(config-sbc-sbe-subscriber-delegate-prof)# duration 100</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td></td>
</tr>
<tr>
<td><code>retry-count #times to retry</code></td>
<td>Configures the number of times the SBC repeats the delegate registration processing after the retry interval ends. The default is 3 times. The range is 0 to 255 times.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>Router(config-sbc-sbe-subscriber-delegate-prof)# retry-count 5</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td></td>
</tr>
<tr>
<td><code>retry-interval retry time in secs</code></td>
<td>Configures the length of time the SBC waits before it retries delegate registration.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>Router(config-sbc-sbe-subscriber-delegate-prof)# retry-interval 60</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td></td>
</tr>
<tr>
<td><code>refresh-buffer timeout in secs</code></td>
<td>Configures the length of time by which the SBC attempts to renew or refresh the address location with a delegate registration before the specified expiration time (duration). The default is 30 seconds. The range is 1 to 2,147,483 seconds.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>Router(config-sbc-sbe-subscriber-delegate-prof)# refresh-buffer 200</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td></td>
</tr>
<tr>
<td><code>exit</code></td>
<td>Exits Subscriber Delegate Profile configuration mode and enters SBE configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>Router(config-sbc-sbe-del-prof)# exit</code></td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action | Purpose
--- | ---
**Step 10** | **subscriber aor**
**Example:**
Router(config-sbc-sbe)# subscriber sip:bob@isp.example
Configures a delegate registration for a specified subscriber associated with an individual client device. Enters into subscriber-entry configuration mode where SIP contact info can be configured for the delegate registration.

**Step 11** | **sip-contact uri**
**Example:**
Router(config-sbc-sbe-subscriber-entry)# sip-contact sip:steve@10.1.1.2
Configures the SIP contact information for a specified URI IP address location or address of record. Enters into subscriber-contact (SIP) configuration mode. The URI is a string of 62 maximum character length.

**Step 12** | **adjacency adjacency name**
**Example:**
Router(config-sbc-sbe-subscriber-contact)# adjacency CallMgrB
Configures the mandatory local subscriber adjacency name of the configured SIP contact.

**Step 13** | **exit**
**Example:**
Router(config-sbc-sbe-subscriber-contact)# exit
Exits subscriber-contact (SIP) configuration mode and enters into subscriber-entry configuration mode to configure delegate registration.

**Step 14** | **delegate-registration hostname**
**Example:**
Router(config-sbc-sbe-subscriber-entry)# delegate-registration sip:registrar@1.1.1.1
Configures a delegate registration for a specified client device or delegate client, and enters into subscriber-delegate configuration mode where registration parameters can be set. The hostname is a string of 64 maximum character length.

**Step 15** | **adjacency adjacency name**
**Example:**
Router(config-sbc-sbe-subscriber-delegate)# adjacency CallMgrA
Configures the adjacency facing the registrar.

**Step 16** | **profile profile name**
**Example:**
Router(config-sbc-sbe-subscriber-delegate)# profile my profile
Applies the delegate registration profile, created previously with the `delegate-profile` command, to a delegate registration subscriber.

**Step 17** | **activate**
**Example:**
Router(config-sbc-sbe-subscriber-delegate)# activate
(Required) Activates the delegate registration.

**Step 18** | **end**
**Example:**
Router(config-sbc-sbe-subscriber-delegate)# end
Exits subscriber-delegate configuration mode and returns to Privileged EXEC mode.
Step 19: `show sbc sbc name sbe sip subscribers delegate`

**Example:**
Router# show sbc mySBC sbe sip subscribers delegate

Displays subscribers for whom delegate registration has been configured. The `delegate` keyword displays the associated URI contact information for subscribers.

Step 20: `show sbc sbc name sbe sip delegate-profiles`

**Example:**
Router# show sbc mySBC sbe sip delegate-profiles

Displays subscriber profiles for subscribers for whom delegate registration has been configured.

### Configuration Examples

This section has the following configuration examples:

- **SIP Fast Registration Example**, page 22-30
- **SoftSwitch Shielding and Aggregate Registration Configuration Examples**, page 22-31
- **Registration Monitoring Examples**, page 22-35
- **Provisional Delegate Registration Examples**, page 22-36
- **Contact Username Passthrough Examples**, page 22-38
- **Alternative Contact Rewriting Example**, page 22-39

### SIP Fast Registration Example

Use the `show platform hardware qfp active feature sbc sfx` command to display the QFP SIP Fast-Register (SFX) counters. Information about how SIP fast-register (SFX) messages are processed, including which SIP REGISTER request packets are punted to the Route Processor (RP) or dropped, may help explain why call rates are low and why the RP CPU load is high.

The following example shows information about the parsing of SIP fast-register (SFX) messages in the Cisco QuantumFlow Processor (QFP):

**Example:**
Router# show platform hardware qfp active feature sbc sfx global

```
SBC QFP SIP Fast Register Data Plane Information

SIP 200 OK Replies generated = 0
SIP REGISTER punts :
  No table entry = 0
  Fast Timer expiry = 0
  Expires=0 = 0
  SIP Syntax Error = 0
  QFP Out of Resources = 0
  QFP Internal Error = 0
SIP REGISTER drops :
  QFP Internal Error = 0
  UDP Length Error = 0
  UDP Checksum Error = 0
```
SoftSwitch Shielding and Aggregate Registration Configuration Examples

The following is a configuration example showing that Aggregate Registration and SoftSwitch Shielding are configured:

```
sbc test
sbe
sip header-profile myheader
header P-Called-Party-ID entry 1
action pass
adjacency sip sippa
header-profile inbound myheader
header-profile outbound myheader
inherit profile preset-access
preferred-transport udp
signaling-address ipv4 99.99.103.150
signaling-port 5080
remote-address ipv4 100.100.1.64 255.255.255.255
signaling-peer 100.100.1.64
signaling-peer-port 5080
registration rewrite-register
account sipp-a
registration aggregate
fast-register disable
header-name to passthrough
request-line rewrite
attach
adjacency sip sipb
nat force-off
header-profile inbound myheader
header-profile outbound myheader
inherit profile preset-core
preferred-transport udp
signaling-address ipv4 99.99.103.150
signaling-port 5082
remote-address ipv4 100.100.1.64 255.255.255.255
signaling-peer 100.100.1.64
signaling-peer-port 5082
account sip-b
registration target address 100.100.1.64
registration target port 5084
fast-register disable
attach
cac-policy-set 1
first-cac-table mytable
first-cac-scope src-adjacency
cac-table mytable
table-type limit adjacency
entry 1
match-value sippa
max-num-calls 10
action cac-complete
complete
active-cac-policy-set 1
call-policy-set 1
first-call-routing-table src-acc-table
first-reg-routing-table src-acc-table
rtg-src-adjacency-table src-acc-table
entry 1
action complete
dst-adjacency sipb
match-adjacency sipa
```

adjacency facing IP-PBX

adjacency facing Registrar
entry 2
  action complete
dst-adjacency sippa
  match-adjacency sippb
  complete
call-policy-set 2
  active-call-policy-set 1
!
vdbe global
  unexpected-source-alerting
media-address ipv4 99.99.103.156
media-timeout 9999
activate
!
Softswitch shielding config
===================
sbc test
sbe
  adjacency sip sippa
    signaling-address ipv4 99.99.103.150
    signaling-port 5080
    remote-address ipv4 100.100.1.64 255.255.255.255
    signaling-peer 100.100.1.64
    signaling-peer-port 5080
    registration rewrite-register
    account sipp-a
    attach
  adjacency sip sippb
    signaling-address ipv4 99.99.103.150
    signaling-port 5082
    remote-address ipv4 100.100.1.64 255.255.255.255
    signaling-peer 100.100.1.64
    signaling-peer-port 5082
    account sipp-b
  registration outgoing timer 86400
  registration target address 100.100.1.64
  registration target port 5084
  attach
call-policy-set 1
  first-call-routing-table src-acc-table
  first-reg-routing-table src-acc-table
  rtg-src-adjacency-table src-acc-table
  entry 1
    action complete
dst-adjacency sippb
    match-adjacency sippa
  entry 2
    action complete
dst-adjacency sippa
    match-adjacency sippb
    complete
  active-call-policy-set 1
!
media-address ipv4 99.99.103.156
media-timeout 9999
activate
!
Router# show sbc test sbe adjacencies sippb detail

SBC Service 'test'
  Adjacency sippb (SIP)
    Status: Attached
    Signaling address: 99.99.103.150:5082
The following example configures SoftSwitch Shielding on adjacency "SoftSwitch:"

```
sbc mySbc
  sbe
  adjacency sip SoftSwitch
    registration outgoing timer <sec>
    registration rewrite-register
    inherit profile preset-core
```

The following example shows detailed output for adjacency SoftSwitch, including the "Register Out Timer:" field that shows the time interval in seconds when the SBC forwards the next REGISTER messages to the softswitch.

```
Router# show sbc mySbc sbe adjacencies SoftSwitch detail
SBC Service "mySbc"
  Adjacency SoftSwitch (SIP)
    Status: Attached
```
Chapter 22  Cisco Unified Border Element (SP Edition) Registration Features

Configuration Examples

Signaling address: 100.100.100.100:5060, VRF Admin
Signaling-peer: 10.10.51.10:5060
Force next hop: No
Account: None
Group: None
In header profile: Default
Out header profile: Default
In method profile: Default
Out method profile: Default
In UA option prof: Default
Out UA option prof: Default
In proxy opt prof: Default
Out proxy opt prof: Default
Priority set name: None
Local-id: None
Rewrite REGISTER: Off
Target address: None
Register Out Timer: 36000 seconds
Register Aggregate: Disabled
NAT Status: Auto Detect
Reg-min-expiry: 30 seconds
Fast-register: Enabled
Fast-register-int: 30 seconds
Authenticated mode: None
Authenticated realm: None
Auth. nonce life time: 300 seconds
IMS visited NetID: None
In inherit profile: Default
Force next hop: No
Home network Id: None
UnEncrypt key data: None
SIP passthrough: No
Rewrite from domain: Yes
Rewrite to header: Yes
Media passthrough: No
Preferred transport: UDP
Hunting Triggers: Global Triggers
Redirect mode: Pass-through
Security: Untrusted
Outbound-flood-rate: None
Ping-enabled: No
Signaling Peer Status: Not Tested

The following example enables Aggregate Registration on adjacency Cary-IP-PBX, which has a preset access profile specified because it faces an access device on a UNI network. The last three commands in the configuration, entered in the correct order, enable the aggregate registration call routing to work.

sbc mySbc
sbe
adjacency sip Cary-IP-PBX
registration rewrite-register
inherit profile preset-access
registration aggregate
header-name to passthrough
request-line request-uri rewrite

The following example displays detailed output for adjacency Cary-IP-PBX, including the “Register Aggregate:” field that shows Aggregate Registration is “Enabled.”

Router# show sbc mySbc sbe adjacencies Cary-IP-PBX detail

SBC Service "mySBC"
Adjacency Cary-IP-PBX (SIP)
Status: Attached
Registration Monitoring Examples

The following example shows how monitoring of event subscriptions as a result of registration state changes is enabled:

```
sbc Raleigh-SBC
sbe
  adjacency sip Cary-IP-PBX
  registration monitor
```

The following example displays detailed output for adjacency Cary-IP-PBX, including the “Registration Monitor:” field that shows Registration Monitoring is “Enabled:”

```
Router# show sbc mySBC sbe adjacencies Cary-IP-PBX detail
```

SBC Service "mySbc"
Adjacency Cary-IP-PBX (SIP)
Status: Attached
Signaling address: 100.100.100.100:5060, VRF Admin
Signaling-peer: 10.10.51.10:5060
Force next hop: No
Account: None
Group: None
In header profile: Default
Out header profile: Default
In method profile: Default
Out method profile: Default
In UA option prof: Default
Out UA option prof: Default
In proxy opt prof: Default
Out proxy opt prof: Default
Priority set name: None
Local-id: None
Rewrite REGISTER: Off
Target address: None
Register Out Timer: 1800 seconds
Register Aggregate: Enabled
NAT Status: Auto Detect
Reg-min-expiry: 30 seconds
Fast-register: Enabled
Fast-register-int: 30 seconds
Authenticated mode: None
Authenticated realm: None
Auth. nonce life time: 300 seconds
IMS visited NetID: None
Inherit profile: Default
Force next hop: No
Home network Id: None
UnEncrypt key data: None
SIPI passthrough: No
Rewrite from domain: Yes
Rewrite to header: Yes
Media passthrough: No
Preferred transport: UDP
Hunting Triggers: Global Triggers
Redirect mode: Pass-through
Security: Untrusted
Outbound-flood-rate: None
Ping-enabled: No
Signaling Peer Status: Not Tested
Rewrite Request-uri: Disabled
Registration Monitor: Enabled

Provisional Delegate Registration Examples

The following example configures a delegate registration profile that can be applied to a delegate registration subscriber.

sbc mySbc sbe
delegate-profile my-profile
duration 1000
retry-count 5
retry-interval 60
refresh-buffer 200

The following example configures a SIP contact for a subscriber, for whom a subscriber detail table exists, and for whom, after the SIP contact is configured, delegate registration can be configured:

sbc mySbc
The following example configures a delegate registration for a specified client device address location, after the SIP contact information has been configured:

```
sbc mySbc
  sbe
    subscriber sip:bob@isp.example
    sip-contact sip:steve@10.1.1.2
    adjacency CallMgrB
    exit
  delegate-registration sip:registrar@1.1.1.1
    adjacency CallMgrA
    activate
```

The following show example displays subscribers for which delegate registration have been configured. The `delegate` keyword displays the associated URI contact information for subscribers.

```
Router# show sbc mySBC sbe sip subscribers delegate

AOR: sip:steve1.cisco.com
Subscriber Location[s]: sip:contact@cisco.com -> CallMgrC
sip:contact2@cisco.com -> CallMgrD
Registrar adj: CallMgrA
Registrar: sip:myreg@172.18.52.148
Register Duration: 1800
Register Retries: 3
Retry Interval: 30
Refresh Buffer: 30
Time left: 0 days
```

The following show example displays subscriber profiles for subscribers for whom delegate registration has been configured.

```
Router# show sbc mySBC sbe sip delegate-profiles

Delegate Profiles:
--------------------------------------------------
Profile         = steve
Duration (secs) = 1800
Retry Count     = 3
Retry Interval (secs) = 30
Refresh Buffer (secs) = 30
--------------------------------------------------
```
Contact Username Passthrough Examples

The following is an example with a single contact showing that the username part of the contact is passed through unchanged:

```
adjacency sip SIPP1Reg
  group SIPP1Reg
  inherit profile preset-core
  signaling-address ipv4 192.168.101.1
  statistics-setting summary
  signaling-port 5060 5062
  remote-address ipv4 192.168.101.12 255.255.255.255
  signaling-peer 192.168.101.12
  signaling-peer-port 7068
  registration target address 192.168.101.12
  registration target port 7069
  registration contact username passthrough
  attach
```

The following is an example flow of multiple registrations for the same subscriber; the example illustrates how a sequence of REGISTER requests registering multiple contacts behaves. This example assumes all headers are omitted from the requests, apart from Contact headers, and that the registrar-facing adjacency has a signaling-port range from 5060 to 5063 (this means 4 local ports are available).

---

**Step 1**
A REGISTER is received registering two contact addresses for the number 5551234.

```
REGISTER sip:5551234@1.2.3.4 SIP/2.0
Contact: <sip:bob@1.1.1.1>
Contact: <sip:robert@1.1.1.1>
```

**Step 2**
The SBC forwards this REGISTER to the registrar having rewritten the contact address and port.

```
REGISTER sip:5551234@1.2.3.4 SIP/2.0
Contact: <sip:bob@192.168.101.1:5060>
Contact: <sip:robert@192.168.101.1:5061>
```

**Step 3**
Another REGISTER is received for the number 5551234, registering another endpoint with a duplicate username of “bob.”

```
REGISTER sip:5551234@1.2.3.4 SIP/2.0
Contact: <sip:bob@2.2.2.2>
```

**Step 4**
The SBC forwards this to the registrar, again passing the username through unchanged.

```
REGISTER sip:5551234@1.2.3.4 SIP/2.0
Contact: <sip:bob@192.168.101.1:5062>
```

**Step 5**
A third endpoint registers for the same number. This endpoint provides a very long contact name in the contact field.

```
REGISTER sip:5551234@1.2.3.4 SIP/2.0
Contact: <sip:this_is_an_extremely_long_contact_username@2.2.2.2>
```

**Step 6**
The SBC forwards this request to the registrar and rewrites the username because it is over the maximum passthrough length (32).

```
REGISTER sip:5551234@1.2.3.4 SIP/2.0
Contact: <sip: 6e83bca53a48bd629a153a93ff8f4af10192.168.101.1:5063>
```
Alternative Contact Rewriting Example

The following example shows how to configure the Alternative Contact Rewriting feature on the SBC:

```
sbc test
  sbe
    adjacency sip core-side-1
      force-signaling-peer
      nat force-off
      inherit profile preset-core
      signaling-address ipv4 9.9.9.1
      remote-address ipv4 10.0.49.78 255.255.255.255
      signaling-peer 10.0.49.78
      registration target address 10.0.49.78
    registration target port 5060
  registration contact username rewrite userid-and-numeric
  attach

    adjacency sip core-side-2
      force-signaling-peer
      nat force-off
      inherit profile preset-core
      signaling-address ipv4 9.9.9.2
      remote-address ipv4 10.0.49.76 255.255.255.255
      signaling-peer 10.0.49.76
      registration target address 10.0.49.76
    registration target port 5060
  registration contact username rewrite numeric
  attach
```

Registering with Softswitch via Cisco SRP Integrated Access Device (IAD) Examples

The following example shows how to register a subscriber with Softswitch, using the Cisco SRP Integrated Access Device (IAD).

This example uses IP realms in the adjacency. See IP Realm Support for information on IP Realms.

```
sbc interop
  sbe
    adjacency sip srp
      nat force-on
      inherit profile preset-access
      preferred-transport udp
      signaling-address ipv4 10.3.127.1
      statistics method summary
      signaling-peer 0.0.0.0
      registration rewrite-register
      realm customer.com
      attach
    adjacency sip meta
      nat force-off
      inherit profile preset-core
      preferred-transport udp
      signaling-address ipv4 99.109.206.106
      statistics method summary
      signaling-peer sig.trav.demo.softswitch.com
      registration target address sig.trav.demo.softswitch.com
      registration target port 5060
```
fast-register disable
header-name To passthrough
header-name From passthrough
realm sig.trav.demo.softswitch.com
attach
call-policy-set 1
first-call-routing-table ctabl
first-reg-routing-table ctabl
rtg-src-adjacency-table ctabl
entry 1
  action complete
dst-adjacency meta
match-adjacency srp
entry 2
  action complete
dst-adjacency srp
match-adjacency meta
complete
active-call-policy-set 1
!
!
!
media-address ipv4 10.3.127.1 realm customer.com
media-address ipv4 99.109.206.106 realm sig.trav.demo.softswitch.com
activate
!

SIP Message Manipulation

You can configure the Cisco Unified Border Element (SP Edition) to selectively examine and manipulate incoming SIP messages on an adjacency.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).

For a complete description of the commands used in this chapter, refer to Cisco Unified Border Element (SP Edition) Command Reference: Unified Model at:

For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

Feature History for SIP Header Manipulation on Cisco Unified Border Element (SP Edition)

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 2.4</td>
<td>The SIP Header Profile, SIP Method Profile, Parameter Profile, Response Code Mapping, SIP Header Manipulation, and Provisional Response filtering features were introduced on Cisco IOS XR along with support for the unified model.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 2.5</td>
<td>The following features were introduced on the Cisco ASR 1000 Series Routers:</td>
</tr>
<tr>
<td></td>
<td>• Ability to Insert Firewall Parameter in SIP Contact Header.</td>
</tr>
<tr>
<td></td>
<td>• Enhanced SIP header manipulation functionality on the Cisco ASR 1000 Series Routers.</td>
</tr>
<tr>
<td></td>
<td>• P-KT-UE-IP header (type of private header) support as part of SIP header manipulation functionality.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 2.6</td>
<td>The following SIP header manipulation functions were enabled with new CLIs on the Cisco ASR 1000 Series Routers:</td>
</tr>
<tr>
<td></td>
<td>• Parse User Name Parameters</td>
</tr>
<tr>
<td></td>
<td>• Suppress Expires Header</td>
</tr>
<tr>
<td></td>
<td>• Configuring Customer P-Asserted-Identity</td>
</tr>
<tr>
<td>Cisco IOS XE Release 3.1S</td>
<td>The following features were added on the Cisco ASR 1000 Series Routers:</td>
</tr>
<tr>
<td></td>
<td>• SIP Destination ID</td>
</tr>
<tr>
<td></td>
<td>• SIP Source ID</td>
</tr>
</tbody>
</table>
Cisco IOS XE Release 3.2S  SBC supports call-policy routing of calls using the hostname in the Request URI. The calls are now routed even in the absence of username in the Request URI.

The Event Header in Publish Method feature was added on the Cisco ASR 1000 Series Routers.

Cisco IOS XE Release 3.3S  The SIP Message Editing feature was added.

Cisco IOS XE Release 3.4S  The SDP Editing Using Script-Based Editors feature was added.

Contents

This chapter contains the following sections:

- SIP Message Editing Using Profiles, page 23-3
- SIP Message Editing Using Editors, page 23-64
- SDP Editing Using Script-Based Editors, page 23-84
SIP Message Editing Using Profiles

This section contains the following information on SIP profiles:

- Information About SIP Profiles, page 23-3
- Method Profiles, page 23-4
- Response Code Mapping, page 23-12
- Header Profiles, page 23-16
- Provisional Response Filtering, page 23-33
- Parameter Profiles, page 23-36
- Ability to Insert Firewall Parameter in the SIP Contact Header, page 23-42
- Configuration Examples for SIP Profiles, page 23-46

Note

From Release 3.3S, the concept of editors has been introduced. An editor is the enhanced version of its corresponding profile. From SIP Message Editing Using Editors? section on page 23-64, all occurrences of profile have been replaced by editor. For example, a method profile is called a method editor.

Information About SIP Profiles

Cisco Unified Border Element (SP Edition) can manipulate the following SIP profiles:

- Method profiles
- Header profiles
- Parameter profiles

Method profiles allow the association of header profiles and parameter profiles to method elements contained in the method profile. You can use actions with method profiles to allow the whitelist to contain blacklisted headers and the blacklist to contain whitelisted headers as well as to reject non-vital methods. This allows any profile to contain mixed actions per-profile.

Header profiles allow complex header manipulation to occur, over and above the existing whitelist and blacklist functionality using actions based on conditional expressions.

Header profiles additionally allow the association of parameter profiles in header elements contained in the profile.

You can use variables to store header content; you can then optionally reconstruct the headers using previously stored variables. You can also match headers based on regular expression matching. You can use conditional matching to match against adjacency settings, transport addresses, and a number of boolean match criteria. You can also use header profiles to reference and make limited modifications to the Request Line.

A header profile can conditionally match any part of a header, but can only replace the entire header. SIP parameter profiles extend this capability to allow changes to be made to individual SIP Request Uniform Resource Identifier (URI) parameters associated with a header.

Parameter profiles allow the removal, replacement, or addition of specific URI parameters within certain vital headers.
You can also associate parameter profiles with methods in method profiles for the purpose of request-line processing per method only.

You can configure multiple store rules, request-lines, and header entries, each with unique actions and/or conditions under which the action is applied. Figure 23-1 shows the hierarchical association of adjacency, method profiles, header profiles, and parameter profiles. The dotted line shows the deprecated method for parameter profile association to method profiles.

**Figure 23-1  SIP Profiles**

**Method Profiles**

SIP methods can be blacklisted and whitelisted dynamically at run-time during receipt of a message (ingress) and at transmission of a message (egress).

A configured method profile allows two types of method profiles for non-vital requests. These can be blacklist (drop) or whitelist (pass). The whitelist action is considered to be the default type for a method if ‘blacklist’ is not present in the command line.

The method profile will contain a list of methods which are either passed on (whitelist) or dropped (blacklist). A single profile can then be associated with each of the inbound or outbound call sides.

Method profiles can be associated with pre-defined header profiles. In addition, pre-defined parameter profiles can be associated with the Request-line per method.

Method profiles are not allowed to blacklist or whitelist vital methods; however, header profiles and parameter profiles can be associated with vital methods.
Status code mapping can be associated with any method type declared in a method profile such that any response identified with this method can be changed. For example, a 503 response to an INVITE could potentially be changed to a 500 response if appropriate mapping is declared against the INVITE method.

This section contains the following topics:

- Restrictions for Configuring Method Profiles, page 23-5
- Information About Method Profiles, page 23-5
- Configuring Method Profiles, page 23-7
- Unconfiguring Method Profiles, page 23-9
- Applying Method Profiles, page 23-11

**Restrictions for Configuring Method Profiles**

Review the following restrictions for method profiles:

- Any given profile must be exclusively a whitelist or a blacklist.
- Two profiles are applied to process any given SIP message: one inbound and, if permitted through that, one outbound.
- Profiles check only SIP methods in the Request Uniform Resource Identifier (URI)
- SIP requests that are blacklisted and non-essential are rejected as a result of a method profile’s rules. SIP responses are always forwarded.
- Any method unknown to Cisco Unified Border Element (SP Edition) which is forwarded as a result of a profile’s rules does not affect creating or deleting a SIP dialog.
- Methods that are essential to the operation of Cisco Unified Border Element (SP Edition) cannot be blacklisted and are implicitly added to any whitelist.
- Profiles cannot be deleted while they are in active use by at least one adjacency.
- In case of non-Information Management System (IMS) preset, there is a default method profile (sip method-profile default). If configured, the default method profile is attached to the adjacencies for which no explicit user-defined method profiles are configured for both inbound and outbound. The sip method profile default is an empty white-list by itself.

**Information About Method Profiles**

After you configure a profile, you can assign it for a default application. Any SIP adjacency can apply it to signaling for that adjacency.

Profiles are an optional part of the configuration—they do not have to be specified for Cisco Unified Border Element (SP Edition) to operate correctly. The default behavior is that requests with one of the essential methods are processed, and all other requests are rejected.

You can add or remove methods from profiles at any time. Each method can optionally be assigned one of three actions with the `action` command:

- Either `pass` or `reject` the method.
- Use the `as-profile` action to select the default profile blacklist or whitelist.

Profiles cannot be deleted while at least one adjacency is using them. You can see which adjacencies are using a profile by entering the following show commands:
show sbc abc-name abc sip method-profile [profile-name]

or

show sbc abc-name abc sip essential-methods
The following methods are part of the essential method set:

- ACK
- BYE
- CANCEL
- INVITE
- NOTIFY
- PRACK
- REFER
- REGISTER
- SUBSCRIBE

To modify parameters in the request-line, associate a parameter profile with a method profile.

Cisco IOS XE Release 2.4 and later contains the following functionalities:

- Predefined header profiles can be associated with outgoing method profiles.
- Predefined parameter profiles can be associated with the request-line per method.

**Note**  
Header profiles and parameter profiles can be associated with essential methods even though method profiles are not allowed to blacklist/whitelist essential methods.

- Response code mapping can be associated with any method type declared in a method profile so that any response identified with the method can be changed. For example, a 503 response to an INVITE could potentially be changed to a 500 response if appropriate mapping is declared against the INVITE method.

### Configuring Method Profiles

This procedure shows how to configure method profiles.

#### SUMMARY STEPS

1. configure
2. sbc sbc-name
3. sbe
4. sip method-profile profile-name
5. description description
6. blacklist
7. pass-body
8. method name
9. action {as-profile | pass | reject}
10. end
11. show sbc sbc-name sbe sip method-profile [profile-name]
12. show sbc sbc-name sbe sip essential-methods
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc sbc-name</td>
<td>Enters the submode for configuring the method profile. Use the <code>sbc-name</code> argument to define the name of the service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config)# sbc mysbc</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Enters the mode of an SBE entity within an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-sbc)# sbe</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> sip method-profile profile-name</td>
<td>Configures a method profile and enters SIP method profile configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>If you enter the <code>profile-name default</code>, the default profile is configured. This profile is used for all adjacencies that do not have a specific profile configured.</td>
</tr>
<tr>
<td><code>Router(config-sbc-sbe)# sip method-profile profile1</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> description description</td>
<td>Adds a description for the specified profile. The <code>no</code> form of this command removes the description.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>This description is displayed when the <code>show</code> command is used for this profile and is displayed for each profile when displaying a summary of all profiles.</td>
</tr>
<tr>
<td><code>Router(config-sbc-sbe-sip-mth)# description mysbc profile1</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> blacklist</td>
<td>Configures a profile to be a blacklist. The <code>no</code> form of this command configures the profile to be a whitelist.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>By default, profiles are whitelists.</td>
</tr>
<tr>
<td><code>Router(config-sbc-sbe-sip-mth)# blacklist</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> pass-body</td>
<td>Permits message bodies to be passed through for non-vital methods accepted by this profile.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>The <code>no</code> form of this command strips the message body out of any non-vital SIP messages matched by this profile.</td>
</tr>
<tr>
<td><code>Router(config-sbc-sbe-sip-mth)# pass-body</code></td>
<td>Non-vital method is same as non-essential method.</td>
</tr>
<tr>
<td><strong>Step 8</strong> method name</td>
<td>Adds a method with the specified name to the profile. Enters the SBE method profile element configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>This field can be 1 to 32 characters (inclusive) in length and is case-insensitive.</td>
</tr>
<tr>
<td><code>Router(config-sbc-sbe-sip-mth)# method test</code></td>
<td>The <code>no</code> form of this command deletes the method with that name from the profile.</td>
</tr>
</tbody>
</table>
### Unconfiguring Method Profiles

The following example shows the proper sequence for unconfiguring a method profile applied to an adjacency. References to the profile must first be removed from all adjacencies. In this example, only one adjacency refers to the profile.

#### SUMMARY STEPS

1. configure terminal
2. sbc sbc-name
3. sbe
4. adjacency sip adjacency-name
5. no method-profile inbound profile-name
6. exit
7. no sip method-profile profile name
8. end

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 9</strong> action (as-profile</td>
<td>Specifies the action to be performed on the parameter.</td>
</tr>
<tr>
<td>pass</td>
<td>reject)</td>
</tr>
<tr>
<td>Example:</td>
<td>as-profile</td>
</tr>
<tr>
<td>Router(config-sbc-sbe-sip-mth-ele)# action</td>
<td>as-profile — Drops the method.</td>
</tr>
<tr>
<td><strong>Step 10</strong> end</td>
<td>pass — Passes the method.</td>
</tr>
<tr>
<td>Example:</td>
<td>reject — Rejects the method.</td>
</tr>
<tr>
<td>Router(config-sbc-sbe-sip-mth-ele)# end</td>
<td>Exits SBE method profile element configuration mode and returns to Privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Step 11</strong> show sbc sbc-name sbe sip method-profile [profile-name]</td>
<td>Displays details for the method profile with the designated name.</td>
</tr>
<tr>
<td>Example:</td>
<td>Use profile-name default to view the default profile.</td>
</tr>
<tr>
<td>Router# show sbc mysbc sbe sip-method-profile profile1</td>
<td>Displays a list of all configured method profiles if no profile-name is specified.</td>
</tr>
<tr>
<td><strong>Step 12</strong> show sbc sbc-name sbe sip essential-methods</td>
<td>Displays a list of the essential methods.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router# show sbc mysbc sbe sip essential-methods</td>
</tr>
</tbody>
</table>
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc sbc-name</td>
<td>Enters the submode for configuring the method profile.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# sbc mysbc</td>
<td>Use the <code>sbc-name</code> argument to define the name of the service.</td>
</tr>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Enters the mode of an SBE entity within an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> adjacency sip adjacency-name</td>
<td>Enters the mode of an SBE SIP adjacency.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe)# adjacency sip sipadj1</td>
<td>Use the <code>adjacency-name</code> argument to define the name of the service.</td>
</tr>
<tr>
<td><strong>Step 5</strong> no method-profile inbound profile-name</td>
<td>Unconfigures profile1 that was used for inbound signaling on adjacency test.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-adj-sip)# no method-profile inbound profile1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> exit</td>
<td>Exits SBE SIP adjacency configuration mode and enters SBE configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-adj-sip)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> no sip method-profile profile name</td>
<td>The <strong>no</strong> form of this command deletes the method with that name from the profile.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe)# no sip method-profile profile1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> end</td>
<td>Exits the SBE mode and returns to Privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe)# end</td>
<td></td>
</tr>
</tbody>
</table>
Applying Method Profiles

This procedure shows how to apply method profiles.

**SUMMARY STEPS**

1. `configure terminal`
2. `sbc sbc-name`
3. `sbe`
4. `adjacency sip adjacency-name`
5. `method-profile inbound profile-name`
6. `end`
7. `show sbc sbc-name sbe sip method-profile name`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc sbc-name</td>
<td>Enters the mode of an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# sbc mysbc</td>
<td>Use the <code>sbc-name</code> argument to define the name of the service.</td>
</tr>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Enters the mode of an SBE entity within an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> adjacency sip adjacency-name</td>
<td>Enters the mode of an SBE SIP adjacency.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe)# adjacency sip test</td>
<td>Use the <code>adjacency-name</code> argument to define the name of the service.</td>
</tr>
<tr>
<td><strong>Step 5</strong> method-profile inbound profile-name</td>
<td>Sets profile1 to be used for inbound signaling on adjacency test.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-adj-sip)# method-profile inbound profile1</td>
<td>When attaching a method profile to an adjacency, the adjacency must be in the “no attach” state.</td>
</tr>
</tbody>
</table>
Response Code Mapping

Response code mapping provides an ability to manipulate the SIP response codes when the messages traverse the Cisco Unified Border Element (SP Edition). The mapping table is applied to inbound messages received at a SIP adjacency or to responses sent out of a SIP adjacency. The mapping is user-configurable on a per SIP method basis so that each SIP method can be mapped differently. lists the mapping limitations on SIP response code.

<table>
<thead>
<tr>
<th>Response Codes</th>
<th>Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>No mapping allowed</td>
</tr>
<tr>
<td>1xx</td>
<td>Maps to 1yy (not 100)</td>
</tr>
<tr>
<td>2xx</td>
<td>Maps to 2yy</td>
</tr>
<tr>
<td>3xx</td>
<td>Maps to 3yy</td>
</tr>
<tr>
<td>4xx</td>
<td>Maps to 4yy, 5yy, or 6yy</td>
</tr>
<tr>
<td>5xx</td>
<td>Maps to 4yy, 5yy, or 6yy</td>
</tr>
<tr>
<td>6xx</td>
<td>Maps to 4yy, 5yy, or 6yy</td>
</tr>
</tbody>
</table>

Response code mapping allows you to:

- Map a particular response code to a specific response code. For example, you can map 401 to 400, but not to 300. You can map 102 to 101, but not 100.
- Map a group of response codes (defined using a wildcard) to a specific response code. For example, you can map 40X to 400, or map all of 4XX to 400.
- Specify exceptions to the wildcard. For example, mapping 2XX to 201, and mapping 200 to 200.

You can use the map-status-code command to add one of more mappings.

Where configuration causes the response code to be mapped to one that is not defined in RFC 3261, Cisco Unified Border Element (SP Edition) applies the reason phrase "Unrecognized status code."

This section contains the following topics:

- Configuring Response Code Mapping, page 23-13
- Applying Response Code Mapping, page 23-15
Restrictions for Response Code Mapping

The following restrictions apply to Response Code Mapping:

- Response code mapping only covers mapping of SIP response codes. H.323 calls cannot have their response codes mapped.
- Certain messages are processed only by the SIP Transaction Manager; mapping of these messages is not possible. For example, badly formatted messages that cannot be interpreted are responded to directly by the SIP Transaction Manager.
- There is no provision for the mapping of SIP reason phrases. The reason phrase will always match the reason code as defined in RFC 3261. A generic reason phrase is applied when the requested reason code has no corresponding definition in RFC 3261. This phrase is a compile time constant.
- Changing the response code could result in an invalid message (for example, mapping the response code could produce a message with mandatory headers missing). There is no provision to ensure that messages contain headers required by the new response code.
- A maximum of 128 mappings is permitted in each direction per adjacency (128 inbound and 128 outbound mappings).

Configuring Response Code Mapping

This procedure shows how to configure response code mapping.

SUMMARY STEPS

1. configure terminal
2. sbc sbc-name
3. sbe
4. sip method-profile profile-name
5. method name
6. map-status-code
7. range statuscoderange value statuscodevalue
8. end
9. show sbc sbc-name sbe sip method-profile [profile-name]
10. show sbc sbc-name sbe sip essential-methods
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>configure terminal</code></td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>sbc sbc-name</code></td>
<td>Enters the submode for configuring the method profile. Use the <code>sbc-name</code> argument to define the name of the service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config)# sbc mysbc</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>sbe</code></td>
<td>Enters the mode of an SBE entity within an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-sbc)# sbe</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> <code>sip method-profile profile-name</code></td>
<td>Configures a method profile. If you enter the <code>profile-name default</code>, the default profile is configured. This profile is used for all adjacencies that do not have a specific profile configured.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-sbc-sbe)# sip method-profile profile1</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> <code>method name</code></td>
<td>Adds a method with the specified name to the profile. This field can be 1 to 32 characters (inclusive) in length and is case-insensitive. The <code>no</code> form of this command deletes the method with that name from the profile.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-sbc-sbe-sip-mth)# method test</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> <code>map-status-code</code></td>
<td>Enters the SIP method profile element configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-sbc-sbe-sip-mth-ele)# map-status-code</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> <code>range statuscoderange value statuscodevalue</code></td>
<td>Maps a range of response codes to a response code.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-sbc-sbe-sip-mth-ele-map)# range 5XX value 500</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> <code>end</code></td>
<td>Exits the method profile mode and returns to Privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-sbc-sbe-sip-mth-prf)# end</code></td>
<td></td>
</tr>
</tbody>
</table>
Chapter 23  SIP Message Manipulation

SIP Message Editing Using Profiles

Applying Response Code Mapping

Apply response code mapping by associating it with an adjacency.

SUMMARY STEPS

1. configure terminal
2. sbc sbc-name
3. sbe
4. adjacency sip adjacency-name
5. method-profile inbound profile-name
6. end
7. show sbc sbc-name sbe sip method-profile name

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc sbc-name</td>
<td>Enters the mode of an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# sbc mysbc</td>
<td>Use the sbc-name argument to define the name of the service.</td>
</tr>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Enters the mode of an SBE entity within an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc)# sbe</td>
<td></td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>Step 4</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>adjacency sip adjacency-name</td>
<td>Enters the mode of an SBE SIP adjacency. Use the adjacency-name argument to define the name of the service.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>Router(config-sbc-sbe)# adjacency sip test</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 5</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>method-profile inbound profile-name</td>
<td>Sets profile1 to be used for inbound signaling on adjacency test.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>Router(config-sbc-sbe-adj-sip)# method-profile inbound profile1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 6</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>end</td>
<td>Exits the header profile mode and returns to Privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>Router(config-sbc-sbe-adj-sip)# end</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 7</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>show sbc sbc-name sbe sip method-profile name</td>
<td>Displays the header profile information.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>Router# show sbc mysbc sbe sip method-profile one</td>
</tr>
</tbody>
</table>

### Header Profiles

Header profiles processing occurs in a two-stage process. In the first stage, the following steps occur:

1. Select next header from the message.
2. Look through the header profile for rules affecting the selected header.
3. In configured order, try to apply each rule to the header.
4. If the action is to add a header, then ignore this rule and move on to the next.
5. If the match condition is FALSE then move onto the next rule, do not evaluate any parameter profile.
6. Apply the action or parameter profile described in the element. If this is to remove the header, then move on to the next header in the message.

The second stage adds new headers to the message. Because it occurs after the first stage, there is a well-defined group of headers in the message. The steps are:

1. Take each rule that adds a header to the message.
2. If the action is to add the first instance of the header only and there is already a header with that name in the message, then move onto the next addition rule.

**Note** If another action has replaced the name of header then it is the replaced name that is used to test whether a new header should be added. That is, any header-name replacements performed in stage 1 are used in this stage of header-name comparisons, and not the original header-names from the arriving message.

3. Add the header if the match condition evaluates to TRUE.
4. Apply any rules defined for that header in user-configured order with this name. Only apply rules that are ordered after the add header rule, if the header was added.
This section contains the following topics:

- Restrictions for Configuring Header Profiles, page 23-17
- Information About Header Profiles, page 23-17
- Header Manipulation, page 23-17
- Header Profile Configuration Information, page 23-25
- Configuring Header Profiles, page 23-25
- Applying Header Profiles, page 23-27

Restrictions for Configuring Header Profiles

Review the following restrictions for header profiles:

- Any given profile must be exclusively a whitelist or a blacklist.
- Two profiles are applied to process any given SIP message: one inbound and, if permitted through that, one outbound.
- SIP headers that are essential to the operation of Cisco Unified Border Element (SP Edition) cannot be blacklisted and are implicitly added to any whitelist.
- Profiles can not be removed while they are in active use by an adjacency.
- For provisional filtering, provisional responses may not be blocked where the sender has required reliable provisional responses (SIP 100rel). This is to ensure that Cisco Unified Border Element (SP Edition) does not interfere with the call setup (as per RFC3262) by dropping the provisional response.
- Header profile conditional matching can be performed against any part of the message. The matches can be exact matches or even sub-strings of any given field.
- The conditions may be associated with a specific header referenced by the header profile header definition, but can also reference other non-vital parts of the message in order to evaluate the conditional expression; thus the condition could be associated with header P-Asserted-Identity while checking against the contents of the Call-Info header.

Information About Header Profiles

After you configure a profile, you can assign it for a default application. Any SIP adjacency can apply it to signaling for that adjacency.

You can add or remove headers from profiles at any time. Headers configured on a profile must contain characters that are valid for a SIP header.

Profiles cannot be deleted while any adjacency is using them. You can see which adjacencies are using a profile by entering the following show command:

```
show sbc sbc-name sbe sip method-profile [profile-name]
```

or

```
show sbc sbc-name sbe sip essential-methods
```

The following are the essential SIP headers, which must not be configured on any profile:

- Allow
- Authorization
- Call-ID
SIP Message Editing Using Profiles

Chapter 23      SIP Message Manipulation

- Contact
- Content-Length
- Content-Type
- CSeq
- Event
- Expires
- From
- Max-Forwards
- Min-Expires
- Proxy-Authenticate
- Proxy-Authorization
- Proxy-Require
- Record-Route
- Referred-By
- Referred-To
- Replaces
- Require
- Route
- Subscription-State
- Supported
- To
- Via
- WWW-Authenticate

Note
Profiles are an optional part of the configuration. If no profile is applicable to a given SIP signal, then the essential headers are processed and all other headers are not forwarded.

Header Manipulation

You can modify non-essential headers in SIP messages using header and parameter profiles. The following information summarizes the supported actions:

- Pass the header unchanged (whitelist functionality).
- Conditionally pass the header unchanged.
- Remove the header (blacklist functionality).
- Conditionally remove the header.
- Replace the name of the header. The replacement name cannot be that of a vital header.
- Conditionally replace the header content (appearing after the “:”).
- Add a new instance of a header to a message regardless of whether or not the header already exists.
- Add the first instance of the header to the message, if a header with this name does not already exist.
A combination of the above actions can be specified as a set or group of actions to be performed within a profile.

The header profiles can be used in method profiles to allow header actions only associated with specific requests types.

Parameter profiles can be associated with headers in header profiles.

Header content can be stored in variables and later expanded during replace-value actions.

Privacy headers are treated as unknown headers, which by default would be blacklisted (stripped). However, the SBC can be configured to pass through SIP Privacy headers.

Regular expression matching can be performed on headers.

You can match against any part of a header but only replace the entire header. A parameter profile extends this capability to change individual SIP URI parameters associated with a header. Header profiles can only modify non-vital header information. To display the vital header information, use the show sbc test sbe sip essential-method, show sbc test sbe sip essential-headers, or show sbc test sbe sip essential-parameters commands.

Parameter profiles can be specified to match the following parts of the message.

- Request URI
- To
- From
- Contact

To modify the parameters in the Request-line, associate a parameter profile with a method profile. To modify the parameters in the Contact, To, or From headers, associate a parameter profile in the header profile.

**Event Header in Publish Method**

As per RFC3903, the SIP PUBLISH request must contain an Event header. In releases earlier than Cisco IOS XE Release 3.2S, the SBC could pass through the PUBLISH method using the existing message manipulation framework, but could not pass through the Event header. The effect of this was that attempts to use the PUBLISH services (containing an Event header) through the SBC were blocked.

From Cisco IOS XE Release 3.2S, the SBC can pass through the PUBLISH method containing Event header using the existing message manipulation framework. Preset header manipulations accessed by inherit-profiles are modified to pass-on the Event header.

The Event Header in Publish Method feature does not affect the behaviors for SUBSCRIBE, REFER, and NOTIFY methods. Event headers are passed through unchanged. For all the other methods, the Event header is treated generically.

**Header Profile Conditional Matching**

To allow header manipulation, a set of conditions can be specified in order to dictate the rules under which the header actions will be applied. Conditional matching allows comparisons to be performed against any part of the message. The matches can be exact matches or even sub-strings of any given field.

The conditions can be associated with a specific header referenced by the header profile header definition, but equally can also reference other non-vital parts of the message in order to evaluate the conditional expression.
Absence of a condition (conditional expression) implies the condition for the action is always true.

Each condition represents a part of the message to be manipulated, and the operation to be performed. A condition can be defined in the following ways:

\textbf{condition} \textbf{comparison-type} \textbf{operator} \textbf{comparison-value}

or

\textbf{condition} \textbf{boolean-operator} \textbf{operator} \{\texttt{true} | \texttt{false}\}

Example:

\begin{verbatim}
condition header-value contains "Cisco"
condition is-request eq \texttt{true}
\end{verbatim}

Table 23-1 lists the comparison types.

\begin{table}[h]
\centering
\begin{tabular}{|l|l|}
\hline
\textbf{Comparison Type} & \textbf{Description} \\
\hline
status-code & Response code value \\
header-value & Current header content \\
header-name name & Content of a different header \\
variables & Match on variable content \\
adjacency & Match on adjacency settings \\
transport & Match on transport addresses or ports \\
header-uri & Match on parts of the URI (username) \\
request-uri & Match on parts of the request-URI (username) \\
\texttt{word} & Match on static strings \\
\hline
\end{tabular}
\caption{Comparison Types}
\end{table}

Table 23-2 lists the operators.

\begin{table}[h]
\centering
\begin{tabular}{|l|l|}
\hline
\textbf{Operator} & \textbf{Description} \\
\hline
[\texttt{not}] eq & Equals or not equal \\
[\texttt{not}] contains & Contains or does not contain \\
[\texttt{not}] regex-match & Regular expression matching (BRE) \\
\texttt{store-as} & Store rules only \\
\hline
\end{tabular}
\caption{Operators}
\end{table}

Table 23-3 lists the boolean operators.

\begin{table}[h]
\centering
\begin{tabular}{|l|l|}
\hline
\textbf{Boolean Operator} & \textbf{Description} \\
\hline
\texttt{is-sip-uri} & Does the header contain a sip: URI \\
\texttt{is-tel-uri} & Does the header contain a tel: URI \\
\hline
\end{tabular}
\caption{Boolean Operators}
\end{table}


The following restrictions apply for conditional matching:

- Multiple conditional expressions against the same header can be added each containing unique actions and conditions to build complex manipulations.
- Each condition must be entered one at a time. To add a subsequent condition to an existing condition, the condition must begin with “and” or “or”. If the condition does not contain “and” or “or”, it effectively overwrites any conditions already defined.
- If no profile-type is explicitly expressed in the header profile command line definition then the assumed header profile type will be “whitelist”.
- Multiple headers of the same type can be declared in any one profile defining either different action types or conditions.
- Character “*” can be used as a wildcard header, although only one wildcard header entry can be configured per profile.
- Duplicate header names with differing actions or conditions can be identified with the “entry <integer>” parameter in the command line. This can be used for the purposes of editing or deletion of a specific action related to a header. If no “entry” in the command line then it is assumed that the first entry related to the header of this header type is being configured.

**Store Rules Declaration**

The data extracted from headers can be stored into variables. The store rules are defined which are executed prior to any header element actions. Store rules are specialized header elements of the format:

```
Store-Rule:<entry>
```

The store rules contain conditions which allow storage in one of the following two ways:

1. A condition can contain a “store-as” keyword to directly store a string or complete header value into a variable.

   ```
   condition comparison-type store-as variable-name
   Example:
   condition header-value store-as var1
   The content of header-value will be stored into var1.
   ```

2. A regular expression can be applied to a header using keyword “regex-match”. If the regular expression contains one or more (up to five max) sets of escaped parentheses ‘( )’ around specific parts of the regular expression, then if the regular expression successfully matches, the values of each parts of the match grouped by the parentheses are extracted and stored into variables defined in the regex-match keyword arguments.

   ```
   condition comparison-type regex-match [store-as variable-name…(up to 5)]
   Example:
   condition header-name P-Asserted-Identity header-value regex-match
   sip:\(.*\)@\[Cc\]isco.com store-as var1
   ```

**Table 23-3 Boolean Operators**

<table>
<thead>
<tr>
<th>Boolean Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>is-request</td>
<td>Is the message a request</td>
</tr>
<tr>
<td>is-100rel-required</td>
<td>Is the call performing 100rel</td>
</tr>
<tr>
<td>is-defined</td>
<td>Test if a variable is defined</td>
</tr>
</tbody>
</table>
For the complete list of comparison types, operators, and boolean operators, refer Table 23-1, Table 23-2, and Table 23-3.

Extracted variables can later be used in the actions which require values such as replace-value, add-first-header/add-header. Variables are expanded by use of “${var}” format within the replacement string.

**Request Line Modification**

You can perform limited modification to the request-line with action replace-value in header profiles. The use of the request-line forming part of the header profiles is the preferred method for changes (including parameter profiles) to the request-line.

The format of the value used in action replace-value is:

```
sip:user@host[:port]
```

The variables that are already extracted to the store rules can be used in the construction of the Request Line.

Example:

```
"sip:${user}@${host}"
```

Request-line is a specialized header element of the format:

```
Request-URI:<entry>
```

**Note**

Changes to the request-line must meet the SIP RFC 3261 formatting rules, and any host declared in the replacement must be a valid host to the SBC. User configuration cannot pre-screen the configured changes due to the possibility of variables being present in the configured replacement value. It is only at run-time when the actual request-line can be determined, and errors in request-line construction can result in call failures. Extreme care must be taken when using this feature to prevent call failures.

**Parse User Name Parameters**

You can configure the SBC to search and parse SIP and SIPS URIs for user name parameters in messages received on an adjacency. If the SIP and SIPS URIs contain any user name parameters, those parameters are treated as regular URI parameters. This is applicable to SIP and SIPS URIs within the Request URI, and the To and From headers for INVITE requests and out-of-dialog requests.

The following is an example of a URI with a username parameter:

```
"sip:username;cic=1234@host.com;user=phone".
```

Here, ‘cic=1234’ is treated as a URI parameter, such as ‘user=phone’, and the username is taken to be ‘username’, instead of ‘username;cic=1234’

Use the command `uri username parameters parse` to enable parsing.

**Suppress Expires Header**

You can configure the SBC to suppress the Expires Header in the outgoing INVITE requests. Use the command `header-name expires suppress` to remove the Expires Header.
Configuring Customer P-Asserted-Identity

You can configure the SBC to specify a value for the P-Asserted-Identity on the outgoing SIP message. The header is added to all requests and responses except ACK, CANCEL, INFO, PRACK, REGISTER and UPDATE.

Use the `header-name p-asserted-id [header-value [header-value] | assert]` command to specify a value for the P-Asserted-Identity.

SIP Destination ID

This feature is applicable only to the INVITE and non-REGISTER out-of-dialogue requests.

When routing a call, the destination address or called party identity is typically derived from the Request URI. However, there are other headers where this information could potentially be derived from, such as To: or P-Called-Party-ID.

You can define an ordered list of headers that can be used to derive the called party address. The headers can include any non-essential SIP header, or To:, and Request URI. A maximum of ten headers can be configured in a header list. The header with priority 1 is analyzed first, the header with priority 2 is analyzed next, and the header with priority 10 is analyzed last.

The following sections describe how this feature works on incoming and outgoing requests.

Incoming Requests

For incoming requests:

- By default, the SBC extracts the called party identity from either the P-Called-Party-ID: header or from the Request URI.
- If the SBC finds multiple instances of a given header in a received SIP message, the first instance is used for called party identity extraction. If the SBC encounters a syntax error while extracting the identity, the SBC creates a log, and moves to the next header in the priority list.
- If a header is not present in the SIP request, or if a header in the header list contains a SIP URI without a username, the SBC moves to the next header in the header list.
- After all headers have been tried without success, the SBC extracts the called party identity from the Request URI.
- The header list may include the Request URI to enable the SBC to look for the called party identity from the Request URI when it gets to a point where the Request URI is prioritized in the list. If the list contains only the Request URI, the SBC looks at only the Request URI.

Outgoing Requests

By default, the SBC reinserts both the domain and the username from the called party identity back into the SIP header from which the identifier originally came on the inbound side.

Outgoing Request URI:

- If the called party identity was originally extracted from the Request URI, the Request URI is reconstructed using the called party identity.
SIP Message Manipulation

Chapter 23

SIP Message Editing Using Profiles

If the called party identity was originally extracted from another header, the username and domain in the Request URI from the SIP message received are preserved. This is done before any SIP header filtering or other editing function (for example, IP/FQDN URI translation) is applied to the Request URI.

Outgoing To Header or Passed Through Arbitrary Header:

- If the called party identity was originally extracted from a header (rather than the Request URI) and that header has been passed through using the inbound adjacency's header manipulation functionality, the SBC inserts the domain and username back into the header, thereby preserving the scheme, URI parameters, and header parameters that were in the original message. Failures due to corruption of header because of the inbound header filtering configuration are logged by the SBC, but other failures are ignored.

- The called party identity may have been edited by the SBC (for example, as part of Number Manipulation) before being reinserted into the outgoing message. This is done only for the first instance of the header in the outbound SIP request before any outbound header filtering or any other editing is applied to the header. There is no restriction on header filtering. You may configure the header editing rules that may subsequently remove or change the header containing the called party identity.

- We recommend that you configure action pass on the inbound header filter profile for all the headers specified in the header list. These headers can then be filtered by the outbound header filter profile.

To configure the destination address header list, use the `dst-address` and `header-priority` commands.

See the "Configuring an Ordered List of Headers for Deriving the SIP Destination Address? section on page 23-28 for details on configuring header-priority for deriving SIP source ID.

The SBC can be configured to perform conditional matching based on these derived values. See the "Header Profile Conditional Matching? section on page 23-19 for more details.

SIP Source ID

When routing a call, the source number can be analyzed and modified using a call policy. The source address is typically derived from the From: header. There are, however, other headers from where this information could potentially be derived from, such as P-Preferred-Identity, P-Asserted-Identity, Remote-Party-ID.

You can define an ordered list of headers that can be used to derive the called party address. The headers can include any non-essential sip header and the From header. The SIP Source ID feature also enables you to derive the source number from an ordered set of headers for the calls that were either redirected or diverted. A maximum of ten headers can be configured in the header list. The header with priority 1 is analyzed first, header with priority 2 is analyzed next and the header with priority 10 is analyzed last.

To configure the source address header list, use the `src-address` command and the `header-priority` command.

See the "Configuring an Ordered List of Headers for Deriving SIP Source Address? section on page 23-30 for details on configuring header-priority for deriving SIP source ID.

SIP Source ID for Diverted Calls

For diverted calls, you can use the address of the party that diverted the call to derive the source address for source analysis. All the diverted calls contain a Diversion: header that contains the details of the party that diverted the call. The SBC can be configured to enter a list of headers for the diverted calls, from which the source number can be derived.
For Cisco IOS XE Release 3.1.0S, this list can only contain one Diversion: header.

To configure the source address header list, use the `div-address` command and the `header-priority` command.

See the "Configuring an Ordered List of Headers for Deriving SIP Source Address of Diverted Calls?" section on page 23-31 for details on configuring header-priority for deriving SIP source ID for diverted calls.

The SBC can be configured to perform conditional matching based on these derived values. See the "Header Profile Conditional Matching?" section on page 23-19 for more details on conditional matching.

### Header Profile Configuration Information

Consideration needs to be given as to the effect of an action or set of actions in conjunction with the default profile behavior (whitelist/blacklist).

An empty blacklist will effectively try to pass on any non-vital header.

An empty whitelist will effectively drop all non-vital headers.

The behavior becomes more complex when conditions are associated with headers.

It is important to consider what actions are defined on the in-bound side. If an empty whitelist header profile is associated with the in-bound side, then no non-vital headers will be visible at all to the out-bound side, and therefore, actions applied to the out-bound sides profile may appear not to work. You may need to consider adding actions to ‘pass’ a specific header on the in-bound side by adding the header to a whitelist (with action as-profile or pass) or adding the header with action ‘pass’ in a blacklist.

For example, if a header profile is defined as a whitelist (default behavior), and a header action to modify the header-value is inserted with a condition, then the action will be processed if the condition is TRUE and the header modified, but will be ignored if the condition is FALSE.

Because the header is inserted into the whitelist it might well be assumed that it would be passed on unmodified if the condition is FALSE, however, if the condition is FALSE, the action (entry) is ignored, and therefore it is as if the header is not present in the whitelist so the header will not be passed on.

To overcome this, a second entry with action ‘pass’ can be entered; thus if the headers condition is TRUE, the content with be modified, but if the condition is false, it will be ignored and continue to process any other entries. The second entry has an action ‘pass’ and will cause the header to be passed on.

### Configuring Header Profiles

This procedure shows how to configure header profiles.

**SUMMARY STEPS**

1. `configure terminal`
2. `sbc sbc-name`
3. `sbe`
4. `sip header-profile profile-name`
5. `blacklist`
6. `description text`
SIP Message Editing Using Profiles

7. header name [entry number]
8. action {add-first-header | add-header | as-profile | drop-msg | pass | replace-name | replace-value | strip}
9. condition [comparison-type | boolean-operator | operator | comparison-value]
10. end
11. show sbc sbc-name sbc sip header-profile [profile-name]
12. show sbc sbc-name sbc sip essential-headers

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc sbc-name</td>
<td>Enters the submode for configuring the header profile.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Use the sbc-name argument to define the name of the service.</td>
</tr>
<tr>
<td>Router(config)# sbc mysbc</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Enters the mode of an SBE entity within an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> sip header-profile profile-name</td>
<td>Configures a header profile.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>If you enter the profile-name default, the default profile is configured. This profile is used for all adjacencies which do not have a specific profile configured.</td>
</tr>
<tr>
<td>Router(config-sbc-sbe)# sip header-profile profile1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> blacklist</td>
<td>Configures a profile to be a blacklist.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>The no form of this command configures the profile to be a whitelist.</td>
</tr>
<tr>
<td>Router(config-sbc-sbe-sip-hdr)# blacklist</td>
<td><strong>Note</strong> By default, profiles are whitelists.</td>
</tr>
<tr>
<td><strong>Step 6</strong> description text</td>
<td>Adds a description for the specified profile.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>The no form of this command removes the description.</td>
</tr>
<tr>
<td>Router(config-sbc-sbe-sip-hdr)# description blacklist profile</td>
<td>This description is displayed when the show command is used for this profile and is displayed for each profile when displaying a summary of all profiles.</td>
</tr>
<tr>
<td><strong>Step 7</strong> header name [entry number]</td>
<td>header name—Configures the SIP header that will be modified. Enters SBC SBE SIP-HDR-ELE configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>entry number—Specifies which action entry to work on.</td>
</tr>
<tr>
<td>Router(config-sbc-sbe-sip-hdr)# header Organization entry 1</td>
<td></td>
</tr>
</tbody>
</table>
## Command or Action

**Step 8**  
**action** (add-first-header | add-header | as-profile | drop-msg | pass | replace-name | replace-value | strip)

**Purpose**  
 Specifies the type of action to be applied to the header.  
 In the example, the action specified is to conditionally replace the header content with a replace value of XYZcompany.

**Example:**  
Router(config-sbc-sbe-sip-hdr-ele)# action replace-value XYZcompany

**Step 9**  
**condition** [comparison-type | boolean-operator | operator | comparison-value]

**Purpose**  
Specifies the condition to match before taking an action to a SIP message profile. If the condition is met, the action specified in step 8 is performed.

**Example:**  
Router(config-sbc-sbe-sip-hdr-ele-act)# condition header-value ABCcompany

**Step 10**  
**end**

**Purpose**  
Exits the SBC SBE SIP-HDR-ELE configuration mode and returns to Privileged EXEC mode.

**Example:**  
Router(config-sbc-sbe-sip-hdr-ele)# end

**Step 11**  
**show** sbc **sbc-name** sbe sip header-profile **profile-name**

**Purpose**  
Displays details for the header profile with the designated name.  
Use the profile-name **default** to view the default profile.

**Example:**  
Router# show sbc mysbc sbe sip header-profile profile1

**Step 12**  
**show** sbc **sbc-name** sbe sip essential-headers

**Purpose**  
Displays a list of the essential headers.

**Example:**  
Router# show sbc mysbc sbe sip essential-headers

---

### Applying Header Profiles

This procedure shows how to apply header profiles.

### SUMMARY STEPS

1. configure
2. sbc **sbc-name**
3. sbe
4. adjacency sip **adjacency-name**
5. header-profile inbound **profile-name**
6. end
7. show sbc **sbc-name** sbe sip header-profile **profile-name**
Chapter 23      SIP Message Manipulation

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router# configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>sbc sbc-name</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config)# sbc mysbc</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>sbe</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc)# sbe</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>adjacency sip adjacency-name</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe)# adjacency sip sipGW</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>header-profile inbound profile-name</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-adj-sip)# header-profile inbound profile1</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>end</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-adj-sip)# end</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>show sbc sbc-name sbe sip header-profile name</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router# show sbc sbc-name sbe sip header-profile name</td>
</tr>
</tbody>
</table>

Configuring an Ordered List of Headers for Deriving the SIP Destination Address

This task configures a list of headers for deriving SIP destination address.

SUMMARY STEPS

1. configure terminal
2. sbc sbc-name
3. sbe
4. sip header-profile profile-id
5. dst-address
### Chapter 23: SIP Message Manipulation

#### SIP Message Editing Using Profiles

6. `header-prio 1 header-name header-name`
7. `header-prio 2 header-name header-name`
8. `header-prio 3 header-name header-name`
9. `end`
10. `show sbc sbc-name sbe sip header-profile profile-id`

### Detailed Steps

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>configure</code></td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# <code>configure</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>sbc sbc-name</code></td>
<td>Enables entry into the mode of an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# <code>sbc mySbc</code></td>
<td>Use the <code>sbc-name</code> argument to define the name of the SBC.</td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>sbe</code></td>
<td>Enables entry into the mode of an SBE entity within an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc)# <code>sbc mySbc sbe</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> <code>sip header-profile</code></td>
<td>Creates the SIP header profile.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe)# <code>sip header-profile</code> Hprofi</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> <code>dst-address</code></td>
<td>Enables entry into the mode to configure destination address.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-sip-hdr)# <code>dst-address</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> <code>header-prio 1 header-name header-name</code></td>
<td>Configures the header priority, and specifies the header to be used.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-sip-hdr-dst)# <code>header-prio 1 header-name</code> P-Called-Party-ID</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> <code>header-prio 2 header-name header-name</code></td>
<td>Configures the header priority, and specifies the header to be used.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-sip-hdr-dst)# <code>header-prio 2 header-name</code> To</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> <code>header-prio 3 header-name header-name</code></td>
<td>Configures the header priority, and specifies the header to be used.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-sip-hdr-dst)# <code>header-prio 3 header-name</code> Request-uri</td>
<td></td>
</tr>
</tbody>
</table>
### Configuring an Ordered List of Headers for Deriving SIP Source Address

This task configures a list of headers for deriving SIP source address.

#### SUMMARY STEPS

1. `configure terminal`
2. `sbc sbc-name`
3. `sbe`
4. `sip header-profile profile-id`
5. `src-address`
6. `header-prio 1 header-name header-name`
7. `header-prio 2 header-name header-name`
8. `header-prio 3 header-name header-name`
9. `end`
10. `show sbc sbc-name sbe sip header-profile profile-id`

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 9</strong> <code>end</code></td>
<td>Enables exit from the destination address configuration mode, and return to the privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-sip-hdr-dst)# end</td>
</tr>
<tr>
<td><strong>Step 10</strong> <code>show sbc sbc-name sbe sip header-profile profile-id</code></td>
<td>Shows the configuration details of the header profile.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router# show sbc mySbc sbe sip header-profile Hprof1</td>
</tr>
</tbody>
</table>

**Command or Action**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>configure</code></td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router# configure</td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>sbc sbc-name</code></td>
<td>Enables entry into the mode of an SBC service. Use the <code>sbc-name</code> argument to define the name of the SBC.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config)# sbc mySbc</td>
</tr>
</tbody>
</table>
Configuring an Ordered List of Headers for Deriving SIP Source Address of Diverted Calls

This task configures a list of headers for deriving SIP source address of diverted calls.

**SUMMARY STEPS**

1. configure terminal

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Enables entry into the mode of an SBE entity within an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc)# sbe mySbc sbe</td>
</tr>
<tr>
<td><strong>Step 4</strong> sip header-profile</td>
<td>Creates SIP header profile.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe)# sip header-profile Hprof1</td>
</tr>
<tr>
<td><strong>Step 5</strong> src-address</td>
<td>Enables entry into the mode to configure source address.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-sip-hdr)# src-address</td>
</tr>
<tr>
<td><strong>Step 6</strong> header-prio 1 header-name header-name</td>
<td>Configures the header priority, and specifies the header to be used.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-sip-hdr-src)# header-prio 1 header-name P-Asserted-Identity</td>
</tr>
<tr>
<td><strong>Step 7</strong> header-prio 2 header-name header-name</td>
<td>Configures the header priority, and specifies the header to be used.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-sip-hdr-src)# header-prio 2 header-name P-Preferred-Identity</td>
</tr>
<tr>
<td><strong>Step 8</strong> header-prio 3 header-name header-name</td>
<td>Configures the header priority, and specifies the header to be used.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-sip-hdr-src)# header-prio 3 header-name From</td>
</tr>
<tr>
<td><strong>Step 9</strong> end</td>
<td>Enables exit from the source address configuration mode and return to the privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-sip-hdr-src)# end</td>
</tr>
<tr>
<td><strong>Step 10</strong> show sbc sbc-name sbc sip header-profile profile-id</td>
<td>Shows the configuration details of the header profile.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router# show sbc mySbc sbc sip header-profile Hprof1</td>
</tr>
</tbody>
</table>
23-32

Chapter 23: SIP Message Manipulation

SIP Message Editing Using Profiles

2. `sbc sbc-name`
3. `sbe`
4. `sip header-profile profile-id`
5. `div-address`
6. `header-prio 1 header-name header-name`
7. `end`
8. `show sbc sbc-name sbe sip header-profile profile-id`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router# configure</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc sbc-name</td>
<td>Enables entry into the mode of an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config)# sbc mySbc</code></td>
<td>Use the <code>sbc-name</code> argument to define the name of the sbc.</td>
</tr>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Enables entry into the mode of an SBE entity within an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-sbc)# sbe mySbc sbe</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> sip header-profile</td>
<td>Creates SIP header profile.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-sbc-sbe)# sip header-profile</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> div-address</td>
<td>Enables entry into the mode to configure source address for diverted calls.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-sbc-sbe-sip-hdr)# div-address</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> header-prio 1 header-name header-name</td>
<td>Configures the header priority, and specifies the header to be used.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-sbc-sbe-sip-hdr-src-div)#</code></td>
<td></td>
</tr>
<tr>
<td><code>header-prio 1 header-name Diversion</code></td>
<td></td>
</tr>
</tbody>
</table>
Chapter 23      SIP Message Manipulation

SIP Message Editing Using Profiles

### Step 7

**end**

**Example:**

```
Router(config-sbc-sbe-sip-hdr-src)# end
```

Enables exit from the source address configuration mode and return to privileged EXEC mode.

### Step 8

**show sbc sbc-name sbe sip header-profile profile-id**

**Example:**

```
Router# show sbc mySbc sbe sip header-profile Hprof1
```

Shows the configuration details of the header profile.

Following is an example for the `show` command output after the header list—for destination address, source address, and diversion address—is configured on SBC:

```
ASR-1002#show sbc mine sbe sip header-profile Hprof1
Header profile "Hprof1"
Description:
  Type:       Whitelist
  dst-address: (inbound only)
                header-prio 1 header-name P-Called-ID
                header-prio 1 header-name To
                header-prio 1 header-name Request-uri
  src-address: (inbound only)
                header-prio 1 header-name Remote-Party-ID
                header-prio 2 header-name P-Preferred-Identity
                header-prio 3 header-name From
  div-address (inbound only)
                header-prio 1 Diversion
  store-rules:
                No store-rule entries found.
  request-line:
                No request-line entries found.
  headers:
    test
    entry 1
      description:
      action add-first-header value "cisco"
      condition is-request eq true
  Not in use with any adjacencies
  Not in use with any method-profile

ASR-1002#
```

### Provisional Response Filtering

Provisional response filtering makes it possible to block 1XX responses (except 100) sent by endpoints. When configuring provisional response filtering, keep the following in mind:

- Provisional responses may not be blocked where the sender has required reliable provisional responses (SIP 100rel).
- Dropping responses where 100_rel is required is not recommended. It may prevent call setup since RFC3262 states subsequent responses should not be sent.
A call attempted with the "Required: 100Rel" header in the INVITE will fail when the adjacency is configured with a header profile to drop 183 messages.

This section contains the following topics:
- Provisional Response Filtering Information, page 23-34
- Configuring Provisional Response Filtering, page 23-34
- Applying Provisional Response Filtering, page 23-35

Provisional Response Filtering Information

Provisional response filtering is achieved by the use of the `action drop-msg` command. The action must be associated with the wildcard header action `*`. A condition should be added to match on the specific response code that must be dropped.

**Note**

The header action `*` can only be used one time in a profile.

Configuring Provisional Response Filtering

1. `configure terminal`
2. `sbc sbc-name`
3. `sbe`
4. `sip header-profile profile-name`
5. `header *`
6. `action drop-msg`
7. `condition status-code`
8. `end`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc sbc-name</td>
<td>Enters the submode for configuring the header profile. Use the <code>sbc-name</code> argument to define the name of the service.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config)# sbc mySBC</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

| Step 3 | configure terminal |

**Example:**
```
Router(config-sbc)# sbe
```

Enters the mode of an SBE entity within an SBC service.

---

| Step 4 | sip header-profile profile-name |

**Example:**
```
Router(config-sbc-sbe)# sip header-profile profile1
```

Configures a header profile.

If you enter the `profile-name default`, the default profile is configured. This profile is used for all adjacencies which do not have a specific profile configured.

---

| Step 5 | header * |

**Example:**
```
Router(config-sbc-sbe-sip-hdr)# header *
```

Configures a profile to be a blacklist.

The `no` form of this command configures the profile to be a whitelist.

**Note** By default, profiles are whitelists.

**Note** In order to filter provisional responses always use the asterisk (*) as the header name with the `header` command as shown in the command example.

---

| Step 6 | action drop-msg |

**Example:**
```
Router(config-sbc-sbe-sip-hdr-ele)# action drop-msg
```

Configures the action to take on an element type in a header.

---

| Step 7 | condition status-code |

**Example:**
```
Router(config-sbc-sbe-sip-hdr-ele-act)# condition status-code eq 183
```

Specifies a condition to match before taking an action to a SIP message profile.

---

| Step 8 | end |

**Example:**
```
Router(config-sbc-sbe-sip-hdr-ele-act)# end
```

Returns to privileged EXEC mode.

---

### Applying Provisional Response Filtering

This procedure shows how to apply provisional response filtering.

### SUMMARY STEPS

1. configure terminal
2. sbc sbc-name
3. sbe
4. adjacency sip adjacency-name
5. header-profile inbound profile-name
6. end
7. show sbc sbc-name sbe sip header-profile name

---

**OL-19829-15**

### Parameter Profiles

Parameter profiles allow you to specify specific URI parameter names and allow the removal, replacement, or the addition of specific non-vital URI parameters within certain headers.

The header profile allows potential conditional matching against SIP URI parameters forming part of a limited set of headers. It only allows complete replacement of the header and or content.

The parameter profile will allow actions to be performed only on the SIP URI parameters and not header parameters.

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>enable the global configuration mode.</td>
</tr>
<tr>
<td>configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>enter the mode of an SBC service.</td>
</tr>
<tr>
<td>sbc sbc-name</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# sbc mysbc</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>enter the mode of an SBE entity within an SBC service.</td>
</tr>
<tr>
<td>sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>enter the mode of an SBE SIP adjacency.</td>
</tr>
<tr>
<td>adjacency sip adjacency-name</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe)# adjacency sip sipGW</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>set the inbound header profile.</td>
</tr>
<tr>
<td>header-profile inbound profile-name</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-adj-sip)# header-profile inbound profile1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>exit the SBE SIP adjacency mode and returns to Privileged EXEC mode.</td>
</tr>
<tr>
<td>end</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-adj-sip)# end</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>show details of the specified SIP header profile.</td>
</tr>
<tr>
<td>show sbc sbc-name sbe sip header-profile name</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# show sbc MySbc sbe sip header-profile profile1</td>
<td></td>
</tr>
</tbody>
</table>
This section contains the following topics:

- Restrictions for Configuring Parameter Profiles, page 23-37
- Information About Parameter Profiles, page 23-37
- Configuring Parameter Profiles, page 23-38
- Applying a Parameter Profile to a Header Profile, page 23-39
- Associating with an Adjacency, page 23-41

Restrictions for Configuring Parameter Profiles

Review the following restrictions for parameter profiles:

- A parameter profile is only permitted to act on parameters associated with SIP URIs and not header parameters.
- To prevent call processing failures, actions cannot be performed against vital (essential) parameters.
- Parameter profiles work only on the outbound side.
- Some of the existing adjacency settings may impact the way parameter actions are affected. For example, consider the adjacency setting Rewrite to Header is set as follows:

```
sbc test
sbe
 adjaceny sip <adj name>
 passthrough [to/from]
```

This setting can cause the To: and or From: headers to be passed from inbound to outbound side. The default setting on an adjacency, however, is FALSE (no “passthrough [to/From]” appears in the show run against the adjacency)’ which means that the To: and From: headers are effectively always re-written on the outbound side by default. The impact of this is that parameter profiles actions applied to the inbound sides To: and/or From: headers will be lost on the outbound side unless ‘passthrough [to/from]’ is set in the configuration. Thus the action **add-not-present** can look like it always adds a parameter on the outbound side, even when the parameter is present on the in-bound side.

- If a parameter profile adds a parameter to the request-line, and the To: header does not have setting ‘passthrough to’ set against the adjacency, then the re-writing of the To: header which is typically based on the Request Line, will cause the parameter to also appear in the To: header.
- The content of the Request-line may affect the behavior of parameter profiles attached to method profiles. If the request-line that arrives on the in-bound side of the call directly addresses the address of Cisco Unified Border Element (SP Edition), then effectively any call that originates on the out-bound side requires a new Request Line to be generated. This means that parameters arriving on the in-bound side are effectively lost and can cause the action add-not-present to look like it always adds a parameter.

  If however, the Request Line address the final destination, then the Request Line is effectively passed across to the outbound side and modified as needed. Parameters in this case are visible on the out-bound side.

Information About Parameter Profiles

Parameter profiles form a set of actions that can be performed against any one header or request-line. Parameter profiles can only be specified against the following parts of the message:
• Request URI
• To
• From
• Contact

To modify parameters in Contact, To, or From headers, associate a parameter profile in the header profile.

To modify parameters in the request-line, associate a parameter profile with a method profile.

Note: Parameter profiles can be associated with essential methods even though method profiles are not allowed to blacklist/whitelist essential methods.

Configuring Parameter Profiles

Perform this task to configure parameter profiles.

SUMMARY STEPS

1. configure terminal
2. sbc sbc-name
3. sbe
4. sip parameter-profile {profile-name}
5. parameter {parameter name}
6. action {add-not-present | add-or-replace | strip}
7. end
8. show sbc sbc-name sbe sip parameter-profile [profile name]
9. show sbc sbc name sbe sip essential-parameters

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>configure terminal</td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td>sbc sbc-name</td>
<td>Enters the mode of an SBC service.</td>
</tr>
<tr>
<td>sbe</td>
<td>Use the sbc-name argument to define the name of the service.</td>
</tr>
<tr>
<td>sip parameter-profile {profile-name}</td>
<td></td>
</tr>
<tr>
<td>parameter {parameter name}</td>
<td></td>
</tr>
<tr>
<td>action {add-not-present</td>
<td>add-or-replace</td>
</tr>
<tr>
<td>end</td>
<td></td>
</tr>
<tr>
<td>show sbc sbc-name sbe sip parameter-profile [profile name]</td>
<td></td>
</tr>
<tr>
<td>show sbc sbc name sbe sip essential-parameters</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Step 4</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.</td>
<td><code>sip parameter-profile {profile-name}</code></td>
<td>Configures a parameter profile and enters SBE SIP header configuration mode.</td>
</tr>
</tbody>
</table>

**Example:**
```bash
Router(config-sbc-sbe)# sip parameter-profile parmprof1
```

<table>
<thead>
<tr>
<th>Step 5</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.</td>
<td><code>parameter {parameter name}</code></td>
<td>Adds a parameter with a specified name to the parameter profile.</td>
</tr>
</tbody>
</table>

**Example:**
```bash
Router(config-sbc-sbe-sip-prm)# parameter user
```

<table>
<thead>
<tr>
<th>Step 6</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.</td>
<td>`action {add-not-present</td>
<td>add-or-replace</td>
</tr>
</tbody>
</table>

**Example:**
```bash
Router(config-sbc-sbe-sip-prm-ele)# action add-not-present value phone
```

<table>
<thead>
<tr>
<th>Step 7</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.</td>
<td><code>end</code></td>
<td>Exits the SBE parameter profile parameter configuration mode and returns to Privileged EXEC mode.</td>
</tr>
</tbody>
</table>

**Example:**
```bash
Router(config-sbc-sbe-sip-prm-ele)# end
```

<table>
<thead>
<tr>
<th>Step 8</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.</td>
<td><code>show sbc sbc-name sbe sip-parameter-profile {profile-name}</code></td>
<td>Displays details for the parameter profile with the designated name. Use the name default to view the default profile.</td>
</tr>
</tbody>
</table>

**Example:**
```bash
Router# show sbc mysbc sbe sip parameter-profile profile1
```

<table>
<thead>
<tr>
<th>Step 9</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.</td>
<td><code>show sbc sbc-name sbe sip essential-headers</code></td>
<td>Displays a list of the essential headers.</td>
</tr>
</tbody>
</table>

**Example:**
```bash
Router# show sbc mysbc sbe sip essential-headers
```

### Applying a Parameter Profile to a Header Profile

Perform this task to apply parameter profiles to a header profile.

### SUMMARY STEPS

1. configure terminal
2. `sbc sbc-name`
3. `sbe`
4. `sip header-profile header-profile-name`
5. `header header-name`
6. `parameter-profile parameter-profile-name`
7. `end`
8. `show sbc sbc-name sbe sip header-profile {profile-name}`
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure terminal</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router# configure terminal</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>sbc sbc-name</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config)# sbc mysbc</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>sbe</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc)# sbe</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>sip header-profile header-profile-name</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-sip)# sip header-profile profile1</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>header header-name</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-sip-hdr)# header P-Asserted-Identity</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>parameter-profile parameter-profile-name</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-sip-hdr-ele)# parameter-profile parmprof1</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>end</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-sip-hdr-ele)# end</td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>show sbc sbc-name sbe sip header-profile name</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router# show sbc sbc-name sbe sip header-profile name</td>
</tr>
</tbody>
</table>
Associating with an Adjacency

Perform the following steps to associate a header profile with an adjacency.

SUMMARY STEPS

1. `configure terminal`
2. `sbc sbc-name`
3. `sbe`
4. `adjacency sip adjacency-name`
5. `header-profile inbound profile-name`
6. `end`
7. `show sbc sbc-name sbe sip header-profile name`

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><code>configure terminal</code> Enables the global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router# configure terminal</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><code>sbc sbc-name</code> Enters the mode of an SBC service. Use the <code>sbc-name</code> argument to define the name of the service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config)# sbc mySBC</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><code>sbe</code> Enters the mode of an SBE entity within an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc)# sbe</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td><code>adjacency sip adjacency-name</code> Enters the mode of an SBE SIP adjacency. Use the <code>adjacency-name</code> argument to define the name of the service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe)# adjacency sip sipGW</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td><code>header-profile inbound profile-name</code> Sets profile1 to be used for inbound signaling on adjacency sipGW.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-adj-sip)# header-profile inbound profile1</td>
</tr>
</tbody>
</table>
SIP Message Editing Using Profiles

Chapter 23  SIP Message Manipulation

SIP Message Editing Using Profiles

Ability to Insert Firewall Parameter in the SIP Contact Header

This feature enables Cisco Unified Border Element (SP Edition) to insert the calling party’s network information (IP address) into SIP headers.

You can use this feature to insert the public IP address for user equipment (UE) that is behind the Network Address Translation (NAT) devices into the SIP contact header as a “firewall” parameter. Inserting a firewall parameter in the header is needed because public IP address information in SIP messages is required in order to properly charge the related parties.

A sample modified contact header in SIP message is the following:

```
Contact:<sip:ea7cf5084c04f49e77644dbe53fd5f1d@10.140.90.6;transport=udp;firewall=10.0.48.41>;Expires=600
```

See "Ability to Insert Firewall Parameter in SIP Contact Header Examples?" section on page 23-64 for examples on inserting IP address information into SIP contact headers.

Configuring Ability to Insert Firewall Parameter in the SIP Contact Header

Perform these tasks to configure this feature.

**SUMMARY STEPS**

1. configure terminal
2. sbc sbc-name
3. sbe
4. sip parameter-profile profile-name
5. parameter {parameter name}
6. action {add-not-present [value] {private-ip-address | public-ip-address | access-user-data}| add-or-replace [value] {private-ip-address | public-ip-address | access-user-data}| strip}
7. exit
8. sip parameter-profile profile-name
9. parameter {parameter name}
10. action {add-not-present [value] {private-ip-address | public-ip-address | access-user-data}| add-or-replace [value] {private-ip-address | public-ip-address | access-user-data}| strip}
11. exit

**Example:**

```
Router(config-sbc-sbe-sip-hdr-prf)# end
```

Exits the header profile mode and returns to Privileged EXEC mode.

**Example:**

```
Router# show sbc sbc-name sbe sip header-profile name
```

Displays the header profile information.
## SIP Message Editing Using Profiles

12. `sip header-profile profile-name`
13. `action {add-not-present [value] {private-ip-address | public-ip-address | access-user-data}| add-or-replace [value] {private-ip-address | public-ip-address | access-user-data}| strip}`
14. `exit`
15. `header header-name`
16. `entry entry_num {action [add-header | as-profile | drop-msg | pass | replace-name | replace-value | strip] | parameter-profile name}`
17. `parameter-profile name`
18. `sip header-profile profile-name`
19. `header header-name`
20. `entry entry_num {action [add-header | as-profile | drop-msg | pass | replace-name | replace-value | strip] | parameter-profile name}`
21. `parameter-profile name`

## Detailed Steps

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td></td>
</tr>
<tr>
<td><code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td></td>
</tr>
<tr>
<td><code>sbc sbc-name</code></td>
<td>Enters the configuration mode of an SBC service. Use the <code>sbc-name</code> argument to define the name of the service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config)# sbc mysbc</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td></td>
</tr>
<tr>
<td><code>sbe</code></td>
<td>Enters the configuration mode of the signaling border element (SBE) function of the SBC.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-sbc)# sbe</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td></td>
</tr>
<tr>
<td><code>sip parameter-profile (profile-name)</code></td>
<td>Configures a parameter profile and enters SBE SIP header configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-sbc-sbe)# sip parameter-profile proxy-param</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td></td>
</tr>
<tr>
<td><code>parameter (parameter name)</code></td>
<td>Adds a parameter with a specified name to the parameter profile and enters SIP parameter profile parameter configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-sbc-sbe-sip-prm)# parameter firewall</code></td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>----------------------------------------------------------------------------------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td><strong>Step 6</strong> action (add-not-present [value]) (private-ip-address</td>
<td>public-ip-address</td>
</tr>
<tr>
<td>Example:.Router(config-sbc-sbe-sip-prm-ele)# action-strip</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> exit</td>
<td>Exit SBE parameter profile parameter configuration mode and enters SBE configuration mode.</td>
</tr>
<tr>
<td>Example:.Router(config-sbc-sbe-sip-prm-ele)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> sip parameter-profile {profile-name}</td>
<td>Configure a parameter profile. Enters into SIP parameter profile configuration mode.</td>
</tr>
<tr>
<td>Example:.Router(config-sbc-sbe)# sip parameter-profile access-param</td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong> parameter {parameter name}</td>
<td>Adds a parameter with a specified name to the parameter profile. Enters SIP parameter profile configuration mode.</td>
</tr>
<tr>
<td>Example:.Router(config-sbc-sbe-sip-prm)# parameter firewall</td>
<td></td>
</tr>
<tr>
<td><strong>Step 10</strong> action (add-not-present [value]) (private-ip-address</td>
<td>public-ip-address</td>
</tr>
<tr>
<td>Example:.Router(config-sbc-sbe-sip-hdr-ele)# action add-or-replace value public-ip-address</td>
<td></td>
</tr>
<tr>
<td><strong>Step 11</strong> exit</td>
<td>Exits to SBE configuration mode.</td>
</tr>
<tr>
<td>Example:.Router(config-sbc-sbe-sip-hdr-ele)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 12</strong> sip header-profile profile-name</td>
<td>Configures a header profile. Enters SIP header profile header configuration mode.</td>
</tr>
<tr>
<td>Example:.Router(config-sbc-sbe)# sip header-profile proxy</td>
<td>If you enter the profile-name default, the default profile is configured. This profile is used for all adjacencies which do not have a specific profile configured.</td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>action {add-not-present {value} (\text{private-ip-address</td>
<td>public-ip-address</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-sip-hdr-ele)# action add-or-replace value public-ip-address</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>exit</td>
<td>Exits SBE header profile header configuration mode and enters into SIP header configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-sip-hdr-ele)# exit</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>header name</td>
<td>Configures the profile to contain the header test1. Enters SBE header profile header configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-sip-hdr)# header test1</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>entry entry_num  {action {add-header</td>
<td>as-profile</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-sip-hdr-ele)# entry 1</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>parameter-profile parameter-profile-name</td>
<td>Configures the parameter profile to apply when the header type is matched.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-sip-hdr-ele)# parameter-profile proxy-param</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>sip header-profile profile-name</td>
<td>Configures a header profile. Enters SIP header configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe)# sip header-profile test1</td>
<td>Configures a header profile. Enters SIP header configuration mode. If you enter the profile-name default, the default profile is configured. This profile is used for all adjacencies which do not have a specific profile configured.</td>
</tr>
<tr>
<td>19</td>
<td>header name</td>
<td>Configures the profile to contain the header test1. Enters SBE header profile header configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-sip-hdr)# header test1</td>
<td></td>
</tr>
</tbody>
</table>
SIP Message Manipulation

SIP Message Editing Using Profiles

This section contains the following:

- Method Profile Examples, page 23-46
- Applying Method Profiles Example, page 23-48
- Associating Predefined Header Profiles Example, page 23-48
- Associating Predefined Parameter Profiles Example, page 23-49
- Associating Response Code Mapping Example, page 23-50
- Configuring Header Profiles Example, page 23-50
- Applying Header Profiles Example, page 23-51
- Header Manipulation Examples, page 23-52
- Response Filtering Example, page 23-60
- Parameter Profile Examples, page 23-61
- Ability to Insert Firewall Parameter in SIP Contact Header Examples, page 23-64

Method Profile Examples

The following example shows the commands and output generated when you configure method profiles.

```
Router# configure terminal
Router(config)# sbc umsbc-node3
Router(config-sbc)# sip method-profile test1
Router(config-sbc-sbe)# method abcd
Router(config-sbc-sbe-sip-mth)# blacklist
```

Example:
```
Router(config-sbc-sbe-sip-hdr-ele)# entry 1
action as-profile
```

Configuration Examples for SIP Profiles

Command or Action | Purpose
--- | ---
Step 20 `entry entry_num action [add-header | as-profile | drop-msg | pass | replace-name | replace-value | strip] | parameter-profile name` | Configures an entry in a profile.

Example:
```
Router(config-sbc-sbe-sip-hdr-ele)# entry 1
action as-profile
```

Step 21 `parameter-profile parameter-profile-name` | Configures the parameter profile to apply when the header type is matched.

Example:
```
Router(config-sbc-sbe-sip-hdr-ele)#
parameter-profile access-param
```
This example shows the output for all method profiles.

This command describes the available method profiles which can be used by the adjacencies. By default, the “default” method profile is configured implicitly and applied to both inbound and outbound directions of all the adjacencies. The default method profile is always active unless it is overwritten by a user-configured method profile. “In use” explains whether the method profile is used by any adjacency or not. When the value is Yes, the “default” method profile is applied to all the adjacencies and is in use. However “test1” has been configured, but not applied to any of the adjacencies. Once you apply the test1 method profile to any adjacency, test1 shows Yes in the “In use” field.

```
Router# show sbc test sbe sip method-profile
Method profiles for SBC service "test1"

<table>
<thead>
<tr>
<th>Name</th>
<th>In use</th>
</tr>
</thead>
<tbody>
<tr>
<td>test1</td>
<td>No</td>
</tr>
<tr>
<td>mprof1</td>
<td>No</td>
</tr>
<tr>
<td>default</td>
<td>Yes</td>
</tr>
<tr>
<td>preset-acc-in-mth</td>
<td>No</td>
</tr>
<tr>
<td>preset-std-in-mth</td>
<td>No</td>
</tr>
<tr>
<td>preset-acc-out-mth</td>
<td>No</td>
</tr>
<tr>
<td>preset-core-in-mth</td>
<td>No</td>
</tr>
<tr>
<td>preset-std-out-mth</td>
<td>No</td>
</tr>
<tr>
<td>preset-core-out-mth</td>
<td>No</td>
</tr>
<tr>
<td>preset-ipsec-in-mth</td>
<td>No</td>
</tr>
<tr>
<td>preset-ipsec-out-mth</td>
<td>No</td>
</tr>
<tr>
<td>preset-ibcf-ext-in-mth</td>
<td>No</td>
</tr>
<tr>
<td>preset-ibcf-int-in-mth</td>
<td>No</td>
</tr>
<tr>
<td>preset-ibcf-utr-in-mth</td>
<td>No</td>
</tr>
<tr>
<td>preset-ibcf-int-in-mth</td>
<td>No</td>
</tr>
<tr>
<td>preset-ibcf-utr-out-mth</td>
<td>No</td>
</tr>
</tbody>
</table>
```
This example shows the output for the method profiles test1.

```
Router# show sbc test sbe sip method-profile test1
Method profile "test1"
Description:
Type: Whitelist
Methods:
INVITE
   action as-profile
   map-status-code
   range 50X value 500
   range 60X value 600
   Not in use with any adjacencies
```

### Applying Method Profiles Example

The following examples show the commands and output generated when you are applying a method profile to Cisco Unified Border Element (SP Edition).

The `method-profile inbound test1` command applies method profile “test1” on the inbound direction. It means that for all incoming messages, check for the method type “abcd.” If the “abcd” method arrives, blacklist it and generate error code 405 Method Not Allowed. All other methods are allowed.

```
Router# configure terminal
Router(config)# sbc umsbc-node3
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency sip sipp-10
Router(config-sbc-sbe-adj-sip)# method-profile inbound test1

Router: Nov 13 17:44:28.609 : config[65761]: %MGBL-CONFIG-6-DB_COMMIT: Configuration committed by user 'username'. Use 'show configuration commit changes 1000000297' to view the changes.
Router(config-sbc-sbe-adj-sip)# end
Router: Nov 13 17:44:31.637 : config[65761]: %MGBL-SYS-5-CONFIG_I: Configured from console by username

Router# show sbc umsbc-node3 sbe sip method-profile
Method profiles for SBC service 'umsbc-node3'
Name       In use
===========
test1      Yes
testb      No

Router# show sbc umsbc-node3 sbe sip method-profile test1
Method profile "test1"
Type: Blacklist
Methods:
   abcd
In use by:
   Adjacency: sipp-10 (in)
```

### Associating Predefined Header Profiles Example

This example shows how to ensure that the parameter `myparm=myvalue` is added to the request-line of an INVITE:

First, configure a parameter profile for `myparm`:

```
Router# configure terminal
Router(config)# sbc test
```
SIP Message Manipulation

Chapter 23      SIP Message Manipulation

SIP Message Editing Using Profiles

Router(config-sbc)# sbe
Router(config-sbc-sbe)# sip parameter-profile parmprof1
Router(config-sbc-sbe-sip-prm)# parameter myparm
Router(config-sbc-sbe-sip-prm-ele)# action add-not-present value myvalue

Then configure and associate with a method profile:

Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# sip method-profile mthdprof1
Router(config-sbc-sbe-sip-mth)# method INVITE
Router(config-sbc-sbe-sip-mth-ele)# parameter-profile parmprof1

Finally, associate with an adjacency

Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency sip adj1
Router(config-sbc-sbe-adj-sip)# method-profile outbound mthdprof1

At the inbound side:
INVITE sip:1234567@cisco.com;user=phone SIP/2.0

At the outbound side:
INVITE sip:1234567@cisco.com;user=phone;myparm=myvalue SIP/2.0

Associating Predefined Parameter Profiles Example

The following example shows how to ensure P-Asserted-Identity is always passed in an INVITE if it contains user=phone.

First, configure a header profile which references a P-Asserted-Identity header:

Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# sip header-profile hdrprof1
Router(config-sbc-sbe-sip-hdr)# header P-Asserted-Identity
Router(config-sbc-sbe-sip-hdr-ele)# action pass
Router(config-sbc-sbe-sip-hdr-ele-act)# condition header-value contains user=phone

Then create and associate the header profile with a method profile:

Router(config-sbc-sbe)# sip method-profile mthdprof1
Router(config-sbc-sbe-sip-mth)# method INVITE
Router(config-sbc-sbe-sip-prm-ele)# header-profile hdrprof1
Finally, associate with an adjacency:

```bash
Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency sip adj1
Router(config-sbc-sbe-adj-sip)# method-profile outbound mthdprof1
```

At the inbound side:

```
INVITE sip:1234567@cisco.com;user=phone SIP/2.0
P-Asserted-Identity: "rob" <sip:1234567@cisco.com;user=phone>
```

At the outbound side:

```
INVITE sip:1234567@cisco.com;user=phone SIP/2.0
P-Asserted-Identity: "rob" <sip:1234567@cisco.com;user=phone>
```

### Associating Response Code Mapping Example

The following example shows how to create a status-code map so that all 5XX responses to an INVITE are mapped to 500.

```bash
Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# sip method-profile mthdprof1
Router(config-sbc-sbe-sip-mth)# method INVITE
Router(config-sbc-sbe-sip-mth-ele)# map-status-code
Router(config-sbc-sbe-sip-mth-ele-map)# range 5XX value 500
```

Finally, associate with an adjacency:

```bash
Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency sip adj1
Router(config-sbc-sbe-adj-sip)# method-profile outbound mthdprof
```

At the inbound side:

```
SIP/2.0 501 Not Implemented
```

At the outbound side:

```
SIP/2.0 500 Internal Server Error
```

### Configuring Header Profiles Example

The following example shows the commands and output generated when you configure the header profiles.

```bash
Router(config)# sbc umsbc-node3 sbe
Router(config-sbc-sbe)# sip header-profile EXAMPLE
Router(config-sbc-sbe-sip-hdr)# blacklist
Router(config-sbc-sbe-sip-hdr)# header abcd
Router# show sbc sbc4 sbe sip header-profile EXAMPLE
```

Header profile EXAMPLE
  Type:  Whitelist
  Headers:
  abcd
Cisco-Guid
Entry 1:
  action add-first-header
User-Agent:
Entry 1:
  action as-profile
Remote-Party-ID
Entry 1:
  action strip
  condition header-value contains user=phone
Entry 2:
  parameter-profile adduser
P-Asserted-Identity
Entry 1:
  action strip
  condition header-value contains user=phone
Organisation
Entry 1:
  action replace-value value Cisco-Systems
  condition header-value contains MCI

In use by:
  Adjacency: callgen100sip (in, out)

Applying Header Profiles Example

The following example shows the commands and output generated when you are applying a header profile to Cisco Unified Border Element (SP Edition).

Router# configure terminal
Router(config)# sbc umsbc-node3 sbe
Router(config-sbc-sbe)# adjacency sip sipp-10
Router(config-sbc-sbe-adj-sip)# header-profile inbound test1
Router(config-sbc-sbe-adj-sip)# header-profile outbound test1
Router# show sbc umsbc-node3 sbe sip header-profile test1

Header profile "test1"
  Type:  Blacklist
  Headers:  abcd
  In use by:
    Adjacency: sipp-10 (in, out)

show running-config

sbc umsbc-node3
  sbe
  activate

sip header-profile test1
  blacklist
  header abcd

! adjacency sip sipp-10
  header-profile inbound test1
  header-profile outbound test1
  signaling-address ipv4 88.88.109.8
  signaling-port 5060
  remote-address ipv4 10.10.105.222 255.255.255.255
  security trusted-encrypted
  signaling-peer 10.10.105.222
  signaling-peer-port 5060
account sip-customer

Header Manipulation Examples

Example—Removing P-Asserted-Identity Header

The following example shows how to remove the header in any message if the header P-Asserted-Identity contains user=phone.

First, access the header:

```
Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# sip header-profile headprof1
Router(config-sbc-sbe-hdr)# header P-Asserted-Identity
Router(config-sbc-sbe-hdr-ele)# action strip
Router(config-sbc-sbe-hdr-ele-act)# condition header-value contains user=phone
```

Next, associate the header with an adjacency:

```
Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency sip adj1
Router(config-sbc-sbe-sip)# header-profile outbound headprof1
```

At the inbound side:

P-Asserted-Identity: "rob" <sip:1234567@cisco.com;user=phone>

At the outbound side:

No P-Asserted-Identity header present

Add this condition in addition to a previous existing condition:

```
Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# sip header-profile headprof1
Router(config-sbc-sbe-hdr)# header P-Asserted-Identity
Router(config-sbc-sbe-hdr-ele)# entry 2
Router(config-sbc-sbe-hdr-ele)# action strip
Router(config-sbc-sbe-hdr-ele-act)# condition header-value contains user=phone
```

Finally, associate the header profile with an adjacency:

```
Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency sip adj1
Router(config-sbc-sbe-sip)# header-profile outbound headprof1
```

At the inbound side:

```
INVITE sip:1234567@cisco.com;user=phone SIP/2.0
...
P-Asserted-Identity: "rob" <sip:1234567@cisco.com;user=phone>
```
At the outbound side:

```
INVITE sip:1234567@cisco.com;user=phone SIP/2.0
...
<No P-Asserted-Identity header present>
```

**Example—Removing Header Based on Condition in Another Header**

The next example shows how to remove a header based on a condition in another header in the message. First, strip the P-Asserted-Identity header, but only if Call-Info: contains "telephone-event."

```
Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# sip header-profile headprof1
Router(config-sbc-sbe-hdr)# header P-Asserted-Identity
Router(config-sbc-sbe-hdr-ele)# action strip
Router(config-sbc-sbe-hdr-ele-act)# condition header-name Call-Info header-value contains telephone-event
```

Then associate the header profile with an adjacency:

```
Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency sip adj1
Router(config-sbc-sbe-sip)# header-profile outbound headprof1
```

At the inbound side:

```
INVITE sip:1234567@cisco.com;user=phone SIP/2.0
...
P-Asserted-Identity: "rob" <sip:1234567@cisco.com;user=phone>
...
Call-Info: <sip:8985010.131.132.6>;method="NOTIFY;Event=telephone-event;Duration=1000"
```

The result at the outbound side:

```
INVITE sip:1234567@cisco.com;user=phone SIP/2.0
...
<No P-Asserted-Identity header present>
```

**Example—Removing Organization Header from All Responses**

The next example removes an Organization header from all Responses:

```
Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# sip header-profile headprof1
Router(config-sbc-sbe-hdr)# header Organization
Router(config-sbc-sbe-hdr-ele)# action strip
Router(config-sbc-sbe-hdr-ele-act)# condition status-code eq 200
```

Associate the header profile with an adjacency:

```
Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency sip adj1
Router(config-sbc-sbe-sip)# header-profile outbound headprof1
```
At the inbound side:
SIP/2.0 200 OK
...
Allow: INVITE, ACK, PRACK, SUBSCRIBE, BYE, CANCEL, NOTIFY, INFO, REFER, UPDATE

At the outbound side:
SIP/2.0 200 OK
...
<No allow header present>

Example—Transforming a Header into Another Header

This example transforms one header into another header (Diversion into Hist-Info).

Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# sip header-profile headprof1
Router(config-sbc-sbe-hdr)# header Diversion
Router(config-sbc-sbe-hdr-ele)# action replace-name value Hist-Info

Associate the header profile with an adjacency:

Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency sip adj1
Router(config-sbc-sbe-sip)# header-profile outbound headprof1

At the inbound side:
INVITE sip:1234567@cisco.com;user=phone SIP/2.0
...
Diversion: <sip:1234567@cisco.com>;reason=unconditional;counter=1;privacy=off

At the outbound side:
INVITE sip:1234567@cisco.com;user=phone SIP/2.0
...
Hist-Info: <sip:1234567@cisco.com>;reason=unconditional;counter=1;privacy=off

Example—Outgoing Messages Contain a Specific Header

This example ensures all outgoing messages contain a specific header (Organization: Cisco.com).

Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# sip header-profile headprof1
Router(config-sbc-sbe-hdr)# header Organization
Router(config-sbc-sbe-hdr-ele)# action add-first-header value cisco.com

Associate the header profile with an adjacency:

Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency sip adj1
Router(config-sbc-sbe-sip)# header-profile outbound headprof1

At the inbound side:
INVITE sip:1234567@cisco.com;user=phone SIP/2.0
...<no Organization header present>

At the outbound side:
INVITE sip:1234567@cisco.com;user=phone SIP/2.0
...Organization: cisco.com

Example—Blacklisting a Header

This example blacklists a header (all instances are removed for any method/response).

Note: This can only be performed against a header profile type of blacklist

Router# configure terminal
Router(config)# sbc test
Router(config-sbc) sbe
Router(config-sbc-sbe)# sip header-profile headprof1
Router(config-sbc-sbe-hdr-ele)# blacklist
Router(config-sbc-sbe-sip-hdr)# header Organization

Or:

Router# configure terminal
Router(config)# sbc test
Router(config-sbc) sbe
Router(config-sbc-sbe-hdr)# sip header-profile headprof1
Router(config-sbc-sbe-hdr-ele)# blacklist
Router(config-sbc-sbe-sip-hdr)# header Organization
Router(config-sbc-sbe-sip-hdr)# action as-profile

Associate the header profile with an adjacency:

Router# configure terminal
Router(config)# sbc test
Router(config-sbc) sbe
Router(config-sbc-sbe)# adjacency sip adj1
Router(config-sbc-sbe-sip)# header-profile outbound headprof1

At the inbound side:
INVITE sip:1234567@cisco.com;user=phone SIP/2.0
...Organization: cisco.com

At the outbound side:
INVITE sip:1234567@cisco.com;user=phone SIP/2.0
...<no Organization: header present>
**Example—Whitelisting a Header**

This example whitelists a header (pass in all methods/responses).

**Note** This can only be specified against a whitelist type of profile which is a default profile and same as “no blacklist.”

```bash
Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# sip header-profile headprof1
Router(config-sbc-sbe-hdr)# header Organization
Or:
Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# sip header-profile headprof1
Router(config-sbc-sbe-hdr)# header Organization
Router(config-sbc-sbe-hdr-ele)# action as-profile
```

Associate the header profile with an adjacency:

```bash
Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency sip adj1
Router(config-sbc-sbe-sip)# header-profile outbound headprof1
```

At the inbound side:

```
INVITE sip:1234567@cisco,com;user=phone SIP/2.0
...
Organization: cisco.com
```

At the outbound side:

```
INVITE sip:1234567@cisco,com;user=phone SIP/2.0
...
Organization: cisco.com
```

**Example—Passing a Date Header**

This example passes a header (Date) conditionally in a 200 response.

```bash
Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# sip header-profile headprof1
Router(config-sbc-sbe-hdr)# header Date
Router(config-sbc-sbe-hdr-ele)# action pass
Router(config-sbc-sbe-hdr-ele-act)# condition status-code eq 200
```

Associate with an adjacency:

```bash
Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency sip adj1
Router(config-sbc-sbe-sip)# header-profile outbound headprof1
```
At the inbound side:
Ensure no other responses contain a Date: header
SIP/2.0 200 OK
...  
Date: Mon, 01 Jan 2008 GMT

At the outbound side:
SIP/2.0 200 OK
...  
Date: Mon, 01 Jan 2008 GMT

Also try all responses containing a Date: header and ensure the 200 OK only contains one

**Example—Stripping Organization Headers in INVITE**

This example strips all 'Organization' headers in an INVITE. To do this, a header profile is created and then associated it with a method profile.

```
Note  
Header profiles can be associated with vital (essential) methods.
```

```
Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# sip header-profile headerprof1
Router(config-sbc-sbe-hdr)# blacklist
Router(config-sbc-sbe-hdr-ele)# header Organization

Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# sip method-profile methodprof1
Router(config-sbc-sbe-sip-mth)# blacklist
Router(config-sbc-sbe-sip-mth)# method INVITE
Router(config-sbc-sbe-sip-mth-ele)# header-profile headerprof1

Associate with an adjacency:
```
```
Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency sip adj1
Router(config-sbc-sbe-sip)# method-profile outbound methodprof1

At the inbound side:
INVITE sip:1234567@cisco.com;user=phone SIP/2.0
...  
Organization: cisco.com

At the outbound side:
INVITE sip:1234567@cisco.com;user=phone SIP/2.0
...  
<no Organization: header present>
Example—Applying Parameter Profile

This example applies a parameter profile to add user=phone into the request-line of an INVITE.

Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# sip parameter-profile test
Router(config-sbc-sbe-sip-prm)# parameter user
Router(config-sbc-sbe-sip-prm-ele)# action add-not-present value phone

Associate with a method profile:

Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe) sip method-profile test
Router(config-sbc-sbe-sip-mth)# method INVITE
Router(config-sbc-sbe-sip-mth-ele) parameter-profile test

Associate with an adjacency:

Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency sip adj1
Router(config-sbc-sbe-sip)# method-profile inbound headprof1

At the inbound side:

INVITE sip:1234567@cisco.com SIP/2.0

At the outbound side:

INVITE sip:1234567@cisco.com;user=phone SIP/2.0

Example—Stripping P-Called-Party-Identity

This example shows how to strip the P-Called-Party-Identity and modify the To: header based on its content:

Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# sip header-profile headprof1
Router(config-sbc-sbe-sip-hdr)# store-rule entry 1
Router(config-sbc-sbe-sip-hdr-ele-act)# description "store the P-Called-Party-Identity"
Router(config-sbc-sbe-sip-hdr-ele-act)# condition header-name P-Called-Party-Identity header-value store-as pcpid
Router(config-sbc-sbe-sip-hdr-ele-act)# exit
Router(config-sbc-sbe-sip-hdr)# header P-Called-Party-Identity entry 1
Router(config-sbc-sbe-sip-hdr-ele-act)# action strip
Router(config-sbc-sbe-sip-hdr-ele-act)# exit
Router(config-sbc-sbe-sip-hdr-ele-act)# exit
Router(config-sbc-sbe-sip-hdr)# header To entry 1
Router(config-sbc-sbe-sip-hdr-ele-act)# action replace-value value "${pcpid}"
Router(config-sbc-sbe-sip-hdr-ele-act)# description "replace the To value"
Router(config-sbc-sbe-sip-hdr-ele-act)# condition variable pcpid is-defined eq true
Associate with an outbound adjacency:

Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency sip adj1
Router(config-sbc-sbe-sip)# header-profile outbound headprof1

Replacing Outbound Request Line Example

This example shows how to replace the outbound request-line with host 172.1.1.1 if user = begins with 1234:

Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# sip header-profile headprof1
Router(config-sbc-sbe-sip-hdr)# store-rule entry 1
Router(config-sbc-sbe-sip-hdr-ele-act)# condition request-uri is-sip-uri eq true
Router(config-sbc-sbe-sip-hdr-ele-act)# condition and request-uri sip-uri-user store-as user
Router(config-sbc-sbe-sip-hdr-ele-act)# exit
Router(config-sbc-sbe-sip-hdr)# request-line entry 1
Router(config-sbc-sbe-sip-hdr-ele-act)# action replace-value value "sip:${user}@172.1.1.1"
Router(config-sbc-sbe-sip-hdr-ele-act)# description "convert RPID param into Privacy header value"
Router(config-sbc-sbe-sip-hdr-ele-act)# condition is-request eq true
Router(config-sbc-sbe-sip-hdr-ele-act)# condition and request-uri is-sip-uri eq true
Router(config-sbc-sbe-sip-hdr-ele-act)# condition and request-uri sip-uri-user regex-match "^1234"

Example—P-KT-UE-IP Header Support

The P-KT-UE-IP header is a type of private header that is supported as a type of SIP header manipulation. The examples in this section show how to remove any existing P-KT-UE-IP headers from all received messages and then replace them with a single P-KT-UE-IP header for INVITE and OOD requests. In the examples, the call is placed from adj1 to adj2.

The following shows how to configure a header profile with two entries. The first entry strips the "P-KT-UE-IP" header and the second entry adds the "P-KT-UE-IP" with a value set to the 18-character string ${msg.rmt_ip_addr}.

Router(config-sbc-sbe)# sip header-profile kt
Router(config-sbc-sbe-sip-hdr)# store-rule entry 1
Router(config-sbc-sbe-sip-hdr-ele-act)# condition adjacency signaling-peer store-as address
Router(config-sbc-sbe-sip-hdr-ele-act)# exit
Router(config-sbc-sbe-sip-hdr)# header P-KT-UE-IP
Router(config-sbc-sbe-sip-hdr-ele-act)# entry 1 action strip
Router(config-sbc-sbe-sip-hdr-ele-act)# exit
Router(config-sbc-sbe-sip-hdr-ele-act)# entry 2 action add-header value "${address}"
The following applies the above header profile to the incoming adjacency as an inbound header profile.

Router(config-sbc-sbe)# adjacency sip adj1
Router(config-sbc-sbe-adj-sip)# header-profile inbound kt

The following configures a header profile to allow passthrough of the "P-KT-UE-IP" header.

Router(config-sbc-sbe)# sip header-profile kt-pass
Router(config-sbc-sbe-sip-hdr)# header P-KT-UE-IP
Router(config-sbc-sbe-sip-hdr-ele)# action pass

The following applies the above header profile to the outgoing adjacency as an outbound header profile.

Router(config-sbc-sbe)# adjacency sip adj2
Router(config-sbc-sbe-adj-sip)# header-profile outbound kt-pass

Response Filtering Example

The following example drops SIP 183 provisional responses from a header profile based on matching the header * associated with inbound and outbound adjacencies.

First, create a header profile headprof1 to match on header * and drop the message:

Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# sip header-profile headprof1
Router(config-sbc-sbe-sip-hdr)# header *
Router(config-sbc-sbe-sip-hdr-ele)# action drop-msg
Router(config-sbc-sbe-sip-hdr-ele-act)# condition status-code eq 183

Associate the profile headprof1 to the inbound side of an adjacency:

Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency sip adjacencyA
Router(config-sbc-sbe-adj-sip)# header-profile inbound headerprof1

Associate the profile headprof1 to the inbound and outbound sides of another adjacency:

Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency sip adjacencyB
Router(config-sbc-sbe-adj-sip)# header-profile inbound headerprof1

Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency sip adjacencyB
Router(config-sbc-sbe-adj-sip)# header-profile outbound headerprof1
**Parameter Profile Examples**

This example shows how to add a user=phone parameter into the To: header if one has not already been specified in a header.

```plaintext
Router# configure terminal
Router (config)# sbc test
Router (config-sbc)# sbe
Router (config-sbc-sbe)# sip parameter-profile parmprof1
Router (config-sbc-sbe-sip-prm)# parameter user
Router (config-sbc-sbe-sip-prm-ele)# action add-not-present value phone
```

Now add to a header profile:

```plaintext
Router# configure terminal
Router (config)# sbc test
Router (config-sbc)# sbe
Router (config-sbc-sbe)# sip header-profile headprof1
Router (config-sbc-sbe-sip-hdr)# header To
Router (config-sbc-sbe-sip-hdr-ele)# parameter-profile parmprof1
```

Now associate with an adjacency:

```plaintext
Router# configure terminal
Router (config)# sbc test
Router (config-sbc)# sbe
Router (config-sbc-sbe)# adjacency sip adj1
Router (config-sbc-sbe-sip)# header-profile outbound headprof1
```

At the inbound side:

```plaintext
INVITE sip:1234567@cisco.com;user=phone SIP/2.0
...
To: "rob" <sip:1234567@cisco.com>;tag=1234;
```

At the outbound side:

```plaintext
INVITE sip:1234567@cisco.com;user=phone SIP/2.0
...
To: "rob" <sip:1234567@cisco.com;user=phone>;tag=1234
```

This example removes the 'user' parameter ('user=phone','user=fax' ...) from the To: header.

```plaintext
Router# configure terminal
Router (config)# sbc test
Router (config-sbc)# sbe
Router (config-sbc-sbe)# sip parameter-profile parmprof1
Router (config-sbc-sbe-sip-prm)# parameter user
Router (config-sbc-sbe-sip-prm-ele)# action strip
```

Add to a header profile:

```plaintext
Router# configure terminal
Router (config)# sbc test
Router (config-sbc)# sbe
Router (config-sbc-sbe)# sip header-profile headprof1
Router (config-sbc-sbe-sip-hdr)# header To
Router (config-sbc-sbe-sip-hdr-ele)# parameter-profile parmprof1
```
Finally, associate with an adjacency:

Finally, associate with an adjacency:

```
Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency sip adj1
Router(config-sbc-sbe-sip)# header-profile outbound headprof1
```

At the inbound side:
```
INVITE sip:1234567@cisco.com;user=phone SIP/2.0
...
To: "rob" <sip:1234567@cisco.com;user=phone;tag=1234;
```

At the outbound side:
```
INVITE sip:1234567@cisco.com;user=phone SIP/2.0
...
To: "rob" <sip:1234567@cisco.com>;tag=1234
```

This example shows how to replace 'user=phone' parameter with user=fax or to add user=fax if a user parameter is not present in the header.

```
Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# sip parameter-profile parmprof1
Router(config-sbc-sbe-sipprm)# parameter user
Router(config-sbc-sbe-sipprm-ele)# action add-or-replace value fax
```

Add to a header profile:
```
Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# sip header-profile headprof1
Router(config-sbc-sbe-siphdr)# header To
Router(config-sbc-sbe-siphdr-ele)# parameter-profile parmprof1
```

Finally, associate with an adjacency:
```
Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency sip adj1
Router(config-sbc-sbe-sip)# header-profile outbound headprof1
```

At the inbound side:
```
INVITE sip:1234567@cisco.com;user=phone SIP/2.0
...
To: "rob" <sip:1234567@cisco.com;user=phone;tag=1234;
```

At the outbound side:
```
INVITE sip:1234567@cisco.com;user=phone SIP/2.0
...
To: "rob" <sip:1234567@cisco.com;user=fax>;tag=1234
```

Or
At the inbound side:

INVITE sip:1234567@cisco.com;user=phone SIP/2.0
...  
To: "rob" <sip:1234567@cisco.com;tag=1234; 

At the outbound side:

INVITE sip:1234567@cisco.com;user=phone SIP/2.0
...  
To: "rob" <sip:1234567@cisco.com;user=fax>;tag=1234 

The next example adds 'user=phone' parameter if it is not already present in the header.

Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# sip parameter-profile parmprof1
Router(config-sbc-sbe-sipprm)# parameter user
Router(config-sbc-sbe-sipprm-ele)# action add-not-present value phone 

Add parameter profile to a header profile:

Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# sip header-profile headprof1
Router(config-sbc-sbe-sip hdr)# header To
Router(config-sbc-sbe-sip hdr-ele)# parameter-profile parmprof1 

Finally, associate with an adjacency

Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency sip adj1
Router(config-sbc-sbe-sip)# header-profile outbound headprof1 

At the inbound side:

INVITE sip:1234567@cisco.com;user=phone SIP/2.0
...  
To: "rob" <sip:1234567@cisco.com;user=fax;tag=1234; 

At the outbound side:

No parameter added as a user parameter already exists
INVITE sip:1234567@cisco.com;user=phone SIP/2.0
...  
To: "rob" <sip:1234567@cisco.com>;tag=1234 

Or

At the inbound side:

INVITE sip:1234567@cisco.com;user=phone SIP/2.0
...  
To: "rob" <sip:1234567@cisco.com;tag=1234; 

At the outbound side:

INVITE sip:1234567@cisco.com;user=phone SIP/2.0
...  
To: "rob" <sip:1234567@cisco.com;user=phone>;tag=1234
Ability to Insert Firewall Parameter in SIP Contact Header Examples

This example adds a SIP parameter profile to remove or append the parameter called firewall:

```
Router(config-sbc-sbe)# sip parameter-profile proxy-param
Router(config-sbc-sbe-sip-prm)# parameter firewall
Router(config-sbc-sbe-sip-prm-ele)# action strip
Router(config-sbc-sbe-sip-prm-ele)# sip parameter-profile access-param
Router(config-sbc-sbe-sip-prm)# parameter firewall
Router(config-sbc-sbe-sip-prm-ele)# action add-or-replace value public-ip-address
```

This example adds a SIP header profile and associates the parameter profile with the header profile:

```
Router(config-sbc-sbe-sip-prm-ele)# sip header-profile proxy
Router(config-sbc-sbe-sip-hdr)# header contact entry 1
Router(config-sbc-sbe-sip-hdr-ele)# action as-profile
Router(config-sbc-sbe-sip-hdr-ele)# parameter-profile proxy-param
Router(config-sbc-sbe-sip-hdr-ele)# sip header-profile access
Router(config-sbc-sbe-sip-hdr)# header contact
Router(config-sbc-sbe-sip-hdr-ele)# entry 1 action as-profile
Router(config-sbc-sbe-sip-hdr-ele)# parameter-profile access-param
```

This example adds a SIP header profile to a SIP adjacency:

```
adjacency sip sip-proxy
    header-profile inbound proxy
    header-profile outbound access
    adjacency sip sip-user
    header-profile inbound access
    header-profile outbound proxy
```

SIP Message Editing Using Editors

This section describes body, header, method, option, and parameter editors. The "SDP Editing Using Script-Based Editors? section on page 23-84 describes script-based editors for modifying the SDP content in SIP messages. You can apply any combination of both types of editors on the SBC for editing SIP messages.

In Release 2.4S, profiles were introduced to enable the SBC to conditionally modify SIP messages. You could configure a profile to modify the body, header, method, option, or parameter of SIP messages that met the matching criteria you specified. This approach was flexible but posed the following limitations:

- Matching criteria could not be set for the vital parts of a message because there was a probability of the call failing if the vital parts of the message were modified.
- With certain limited exceptions, the vital parts of a message could not be modified because the original content of these vital parts was not available at the point at which the profiles were applied.

From Release 3.3S, the concept of editors has been introduced. An editor refers to any kind of SBC configuration that is used for conditionally editing SIP messages. Profiles that were introduced in earlier releases are now renamed as editors. For example, body profiles are now known as body editors, header profiles are known as header editors, and so on.

Editors can be associated with an adjacency and linked together so that they can be applied in a specified sequence at run time. In addition, you can test editors by applying them on a test message (a SIP INVITE). You can use the output of the test to determine whether the editors meet your requirements.
In Cisco IOS XE Release 3.3S, the following additional enhancements have been introduced in the SIP Message Editing feature:

- **To and From multimode fiber optic edits**
  Prior to Cisco IOS XE Release 3.3S, the To and From outbound headers of only out-of-dialog messages and dialog-creating messages could be edited. After an edit was performed on a dialog-creating message, the edit was automatically propagated across all the new messages sent on the dialog. From Cisco IOS XE Release 3.3S, edits on the To and From headers can also be performed on in-dialog messages. There is no automatic propagation of these edits. This requires you to ensure that the edits are consistently performed for all messages sent on the dialog.

- **Resource Priority header inspection**
  Prior to Cisco IOS XE Release 3.3S, the Resource Priority header inspection function examined a message before any inbound MMF editing was performed. From Cisco IOS XE Release 3.3S, the Resource Priority header inspection function examines a message after inbound editing has been performed.

- **100rel_required match condition variable**
  Prior to Cisco IOS XE Release 3.3S, the 100rel_required match condition variable was a call property that was updated when new information about 100rel support came in from each call leg. From Cisco IOS XE Release 3.3S, this variable is an indicator of whether the received message is marked as Required: 100rel.

- **Failure responses**
  Prior to Cisco IOS XE Release 3.3S, failures encountered during message editing resulted in the SBC sending a rejection for the unedited message. From Cisco IOS XE Release 3.3S, the response contains the state of the message at the point of failure. For example, headers added during editing are mentioned in the failure response.

The following sections provide information about implementing SIP message editing using body, header, method, option, and parameter editors:

- **Restrictions for SIP Message Editing, page 23-65**
- **Guidelines for Naming Editors, page 23-66**
- **Configuring Editors, page 23-66**
- **Configuration Examples for SIP Message Editors, page 23-76**

### Restrictions for SIP Message Editing

The SIP Message Editing feature does not support the following actions:

- Editing To and From header tags
- Applying the pass and strip actions on To and From header tags
- Outbound editing ofVia headers
- Changing the method types of INVITE, CANCEL, and ACK messages
Guidelines for Naming Editors

Apply the following guidelines while naming an editor:

- Ensure that each editor has a unique name. Apply this guideline across editors. For example, ensure that the name of a header editor is not the same as the name of a method editor.
- Note that an editor and a profile should have the same name to ensure an easy migration path.

Configuring Editors

This task describes how to configure editors on the SBC.

SUMMARY STEPS

1. configure terminal
2. sbc sbc-name
3. sbe
4. sip editor-type {editor | profile}
5. sip body-editor editor-name
6. exit
7. sip method-editor {editor-name | default}
8. exit
9. sip option-editor {editor-name | default}
10. exit
11. sip parameter-editor {editor-name | default}
12. exit
13. sip header-editor {editor-name | default}
14. exit
15. adjacency sip adjacency-name
16. editor-type {editor | profile}
17. header-editor {inbound | outbound} {editor-name | default}
18. method-editor {inbound | outbound} {editor-name | default}
19. option-editor [ua | proxy] {inbound | outbound} {editor-name | default}
20. body-editor {inbound | outbound} {editor-name}
21. editor-list {after-send | before-receive}
22. `editor order-number editor-name [condition [body contains sdp]]`
23. `end`
24. `show sbc sbc-name sbe editors`
25. `show sbc sbc-name sbe sip header-editor [editor-name]`
26. `show sbc sbc-name sbe sip body-editor [editor-name]`
27. `show sbc sbc-name sbe sip method-editor [editor-name]`
28. `show sbc sbc-name sbe sip option-editor [editor-name]`
29. `show sbc sbc-name sbe sip parameter-editor [editor-name]`
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters the global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc sbc-name</td>
<td>Enters the SBC service mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# sbc mysbc</td>
<td><em>sbc-name</em>—Name of the SBC.</td>
</tr>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Enters the SBE configuration mode of the SBC.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> sip editor-type {editor</td>
<td>profile}</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe)# sip editor-type editor</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> sip body-editor editor-name</td>
<td>Creates a body editor to filter non-SDP message bodies from incoming and outgoing SIP messages.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe)# sip body-editor BodyEditor1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> exit</td>
<td>Exits the SIP body configuration mode and enters the SBE configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-mep-bdy)# exit</td>
<td></td>
</tr>
</tbody>
</table>
**Chapter 23  SIP Message Manipulation**

**SIP Message Editing Using Editors**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 7 sip method-editor {editor-name</td>
<td>default}</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe)# sip method-editor</td>
<td>MethodEditor1</td>
</tr>
</tbody>
</table>

- *editor-name*—Specifies the name of the method editor.
- *default*—Configures the default method editor. This editor is used for all the adjacencies that do not have a specific editor configured.

- Enters the SIP method configuration mode. Use the following commands under this mode to configure the method editor:
  - *blacklist*—Sets this editor to be blacklist.
  - *description*—Sets the description for this editor.
  - *method*—Adds a method to this editor.

  The *method* command enters the SIP method editor element configuration mode, where the following commands can be used:
  - *action*—Specifies the action performed on the method.
  - *body-editor*—Adds a body editor to act on the method.
  - *header-editor*—Adds a header editor to act on the method.
  - *map-status-code*—Allows mapping of the response codes received for a method.

| Step 8 exit | Exits the SIP method configuration mode and enters the SBE configuration mode. |
| **Example:**                       |         |
| Router(config-sbc-sbe-mep-mth)# exit |         |

| Step 9 sip option-editor {editor-name | default} | Configures an option editor. |
| **Example:**                       |         |
| Router(config-sbc-sbe)# sip option-editor | OptionEditor1 |    |

- *editor-name*—Specifies the name of the option editor.
- *default*—Configures the default option editor.

- Enters the SIP option configuration mode. Use the following commands under this mode to configure the option editor:
  - *blacklist*—Sets this editor to be blacklist.
  - *description*—Sets the description for this editor.
  - *option*—Adds an option to this editor.

| Step 10 exit | Exits the SIP option configuration mode and enters the SBE configuration mode. |
| **Example:**                       |         |
| Router(config-sbc-sbe-mep-opt)# exit |         |
### Command or Action

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 11</strong> sip parameter-editor editor-name</td>
<td>Configures a parameter editor.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe)# sip parameter-editor ParameterEditor1</td>
<td>Enters the SIP parameter configuration mode. Use the following commands under this mode to configure the parameter editor:</td>
</tr>
<tr>
<td>- blacklist</td>
<td>Sets this editor to be blacklist.</td>
</tr>
<tr>
<td>- description</td>
<td>Sets the description for this editor.</td>
</tr>
<tr>
<td>- parameter</td>
<td>Adds an parameter to this editor.</td>
</tr>
<tr>
<td></td>
<td>The parameter command enters the SIP parameter editor element configuration mode, from where you can configure the action to be taken on an element type in the parameter editor using the action command.</td>
</tr>
</tbody>
</table>

| **Step 12** exit | Exits the SIP parameter configuration mode and enters the SBE configuration mode. |
| **Example:** Router(config-sbc-sbe-mep-prm)# exit | |
| | |
Chapter 23  SIP Message Manipulation


SIP Message Editing Using Editors

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 13</strong> sip header-editor {editor-name</td>
<td>default}</td>
</tr>
<tr>
<td></td>
<td>• editor-name—Specifies the name of the header editor.</td>
</tr>
<tr>
<td></td>
<td>• default—Configures the default header editor.</td>
</tr>
</tbody>
</table>

Example:  
Router(config-sbc-sbe)# sip header-editor HeaderEditor1

Enters the SIP header configuration mode. Use the following commands under this mode to configure the header editor:

• blacklist—Sets this editor to be blacklist.
• description—Sets the description for this editor.
• div-address—Specifies a priority list of headers from which the diverted-by number is to be derived (inbound only). Enters the SIP header editor diversion header configuration mode, from where you can use the following command:
  – header-prio—Specifies a priority-ordered list for extracting the diverted-by address.

• dst-address—Specifies a priority list of headers from which the called party address is to be derived (inbound only). Enters the SIP header editor destination header configuration mode, from where you can use the following command:
  – header-prio—Specifies a priority ordered list for extracting the destination address.

• header—Adds a header to this editor. Enters the SIP header editor header configuration mode, from where you can use the following commands:
  – action—Specifies the type of action. Enters the SIP header editor header action mode, from where you can use the condition command to specify one or more conditions for the action to be effective and the parameter-editor command to specify the parameter editor.
  – parameter-editor—Specifies the parameter editor.
Chapter 23      SIP Message Manipulation

SIP Message Editing Using Editors

Command or Action | Purpose
--- | ---
request-line—Allow actions to modify the Request Line (outbound side only). Enters the SIP header editor header configuration mode, from where you can use the following commands:
  - action—Specifies the type of action. Enters the SIP header editor header action mode, from where you can use the condition command to specify one or more conditions for the action to be effective and the parameter-editor command to specify the parameter editor.
  - parameter-editor—Specifies the parameter editor.
src-address—Specifies a priority list of headers from which the calling party address is to be derived (inbound only). Enters the SIP header editor calling party configuration mode, from where you can use the following command:
  - header-prio—Specifies a priority ordered list for extracting the source address.
store-rule—Creates a store rule to extract variables from headers. Enters the SIP header editor header action configuration mode, from where you can use the following commands:
  - condition—Specifies one or more conditions for the action to be effective.
  - description—Sets the description for this action.

Step 14 | exit
Exits the SIP header configuration mode and enters the SBE configuration mode.

Example:
Router(config-sbc-sbe-mep-hdr)# exit

Step 15 | adjacency sip adjacency-name
Enters the SBE SIP adjacency configuration mode.

Example:
Router(config-sbc-sbe)# adjacency sip SIPP

Step 16 | editor-type {editor | profile}
Specifies the editor type for the SIP adjacency to apply.

Example:
Router(config-sbc-sbe-sip)# editor-type editor

editor—Uses the method, header, option, parameter, or body editor.

profile—Uses the method, header, option, parameter, or body profile.
### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>header-editor (inbound</td>
<td>outbound) {editor-name</td>
</tr>
<tr>
<td></td>
<td>Example: Router(config-sbc-sbe-sip)# header-editor inbound HeaderEditor1</td>
<td><strong>inbound</strong>—Sets the inbound SIP header editor.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>outbound</strong>—Sets the outbound SIP header editor.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>editor-name</strong>—Name of the header editor to be set for inbound or outbound signaling on the adjacency.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>default</strong>—Sets the header editor to the default settings.</td>
</tr>
<tr>
<td>18</td>
<td>method-editor (inbound</td>
<td>outbound) {editor-name</td>
</tr>
<tr>
<td></td>
<td>Example: Router(config-sbc-sbe-sip)# method-editor inbound HeaderEditor1</td>
<td><strong>inbound</strong>—Sets the inbound SIP method editor.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>outbound</strong>—Sets the outbound SIP method editor.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>editor-name</strong>—Name of the method editor to be set for inbound or outbound signaling on the adjacency.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>default</strong>—Sets the method editor to the default settings.</td>
</tr>
<tr>
<td>19</td>
<td>option-editor [ua</td>
<td>proxy] [inbound</td>
</tr>
<tr>
<td></td>
<td>Example: Router(config-sbc-sbe-adj-sip)# option-editor ua inbound OptionHeader1</td>
<td><strong>ua</strong>—Sets the SIP ua option editors.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>proxy</strong>—Sets the SIP proxy option editors.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>inbound</strong>—Sets the inbound SIP option editors.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>outbound</strong>—Sets the outbound SIP option editors.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>editor-name</strong>—Name of editor to use.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>default</strong>—Sets the method editor to the default settings.</td>
</tr>
<tr>
<td>20</td>
<td>body-editor (inbound</td>
<td>outbound) {editor-name}</td>
</tr>
<tr>
<td></td>
<td>Example: Router(config-sbc-sbe-adj-sip)# body-editor inbound BodyEditor1</td>
<td><strong>inbound</strong>—Associates the body editor to act on inbound messages on the SIP adjacency.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>outbound</strong>—Associates the body editor to act on outbound messages on the SIP adjacency.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>editor-name</strong>—Specifies a name for the body editor. The maximum length is 30 characters.</td>
</tr>
</tbody>
</table>

**Note** When the message is passed through the SBC, the body editor is applied in both the inbound and outbound directions on the respective adjacencies on which the message is routed.
### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| 21   | `editor-list (after-send | before-receive)` | Configures a list of registered editors.  
- **after-send**—Specifies that the outgoing message must be edited after it is processed by the adjacency and just before it is forwarded from the adjacency.  
- **before-receive**—Specifies that the incoming message must be edited just after it is received on the adjacency and before the adjacency begins processing it.  
Example:  
```
Router(config-sbc-sbe-adj-sip)# editor-list  
after-send
```
| 22   | `editor order-number editor-name [condition [body contains sdp]]` | Configures an editor in the editor list. For each editor that you want to apply in a sequence, run this command to specify the order of the editor in the editor list.  
**Note** You can add any combination of script-based editors and body, header, method, option, and parameter editors in the editor list.  
- **order-number**—Order in which the editor must be applied. The range is from 1 to 2147483647.  
- **editor-name**—Specifies the name of the editor that you want to apply to messages that are processed by the adjacency.  
- **condition**—Specifies that there are one or more conditions for the editor to be applied.  
- **body contains sdp**—Specifies that the message body must be SDP-based content. The editor is applied only if this condition is met. Include **body contains sdp** in the command for script-based editors.  
Example:  
```
Router(config-sbc-sbe-adj-sip-ed)# editor 1  
bodyeditor1
```
| 23   | `end` | Exits the SIP editor configuration mode, and enters the privileged EXEC mode.  
Example:  
```
Router(config-sbc-sbe-adj-sip)# end
```
| 24   | `show sbc sbc-name sbe editors` | Lists all the configured editors.  
Example:  
```
Router# show sbc mysbc sbe editors
```
### Command or Action

| Step 25 | `show sbc sbc-name sbe sip body-editor [editor-name]` |

**Example:**

Router# show sbc mysbc sbe sip body-editor

BodyEditor1

- **Purpose**: Displays the details of all body editors, or displays details pertaining to the specified body editor.
  - `sbc-name`—Specifies the name of the SBC service.
  - `editor-name`—Specifies the name of the editor and displays details about the specified editor. If omitted, the command shows information about all the SIP body editors.

| Step 26 | `show sbc sbc-name sbe sip header-editor [editor-name]` |

**Example:**

Router# show sbc mysbc sbe sip header-editor

HeaderEditor1

- **Purpose**: Displays the details of all header editors, or displays details pertaining to the specified header editor.
  - `sbc-name`—Specifies the name of the SBC service.
  - `editor-name`—Specifies the name of the editor and displays details about the specified editor. If omitted, the command shows information about all the SIP header editors.

| Step 27 | `show sbc sbc-name sbe sip method-editor [editor-name]` |

**Example:**

Router# show sbc mysbc sbe sip method-editor

MethodEditor1

- **Purpose**: Displays the details of all method editors, or displays details pertaining to the specified method editor.
  - `sbc-name`—Specifies the name of the SBC service.
  - `editor-name`—Specifies the name of the editor and displays details about the specified editor. If omitted, the command shows information about all the SIP method editors.

| Step 28 | `show sbc sbc-name sbe sip option-editor [editor-name]` |

**Example:**

Router# show sbc mysbc sbe sip option-editor

OptionEditor1

- **Purpose**: Displays the details of all option editors, or displays details pertaining to the specified option editor.
  - `sbc-name`—Specifies the name of the SBC service.
  - `editor-name`—Specifies the name of the editor and displays details about the specified editor. If omitted, the command shows information about all the SIP option editors.

| Step 29 | `show sbc sbc-name sbe sip parameter-editor [editor-name]` |

**Example:**

Router# show sbc mysbc sbe sip parameter-editor

ParameterEditor1

- **Purpose**: Displays the details of all parameter editors, or displays details pertaining to the specified parameter editor.
  - `sbc-name`—Specifies the name of the SBC service.
  - `editor-name`—Specifies the name of the editor and displays details about the specified editor. If omitted, the command shows information about all the SIP parameter editors.
Chapter 23      SIP Message Manipulation

Configuration Examples for SIP Message Editors

This section contains the following examples:

- Method Editor Example, page 23-76
- Header Editor Example, page 23-78
- Body Editor Example, page 23-81
- Option Editor Example, page 23-83
- Parameter Editor Example, page 23-83

Method Editor Example

The following example shows how to configure the test1 method editor and the abcd method type on the SBC2 SBC.

Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# sbc SBC2
Router(config-sbc)# sbe
Router(config-sbc-sbe)# sip method-editor test1
Router(config-sbc-sbe-mep-mth)# method abcd
Router(config-sbc-sbe-mep-mth)# blacklist

The following example shows how the show sbc sbe sip method-editor command is used to display details of the meditor1 method editor and the test1 method editor before they have been applied to an adjacency.

Router# show sbc SBC2 sbe sip method-editor meditor1
method-editor "meditor1"
  Description:
  Type:    Whitelist
  Methods:
  INVITE
    action as-editor
    map-status-code
      range 5XX value 500
      range 6XX value 600
  Not in use with any adjacencies

Router# show sbc SBC2 sbe sip method-editor test1
method-editor "test1"
  Description:
  Type:    Blacklist
  Methods:
  abcd
    action as-editor
  Not in use with any adjacencies

The following example shows how the show sbc sbe sip method-editor command is used to display a list of all configured method editors:

Router# show sbc SBC2 sbe sip method-editor
method-editors for SBC service "SBC2"
  Name       In use
  ---------------------------
  test1      No
  meditor1   No
  preset-acc-in-mth     No
### Example—Applying the Method Editor

The `method-editor inbound test1` command applies the test1 method editor on the inbound direction. Therefore, for all incoming messages, the method type abcd is checked. When the abcd method arrives, it is blacklisted and the error code 405 Method Not Allowed is generated. All the other methods are allowed.

```plaintext
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# sbc SBC2
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency sip trans-uac
Router(config-sbc-sbe-adj-sip)# no attach
Router(config-sbc-sbe-adj-sip)# method-editor inbound test1
Router(config-sbc-sbe-adj-sip)# attach
```

The following example shows how the `show sbc sbe sip method-editor` command is used to display details of the test1 method editor after it has been applied to an adjacency.

```plaintext
Router# show sbc SBC2 sbe sip method-editor
method-editors for SBC service "SBC2"

<table>
<thead>
<tr>
<th>Name</th>
<th>In use</th>
</tr>
</thead>
<tbody>
<tr>
<td>test1</td>
<td>Yes</td>
</tr>
<tr>
<td>meditor1</td>
<td>No</td>
</tr>
</tbody>
</table>

Router# show sbc SBC2 sbe sip method-editor test1
method-editor "test1"
    Description:
    Type: Blacklist
    Methods:
    abcd
    action as-editor
    In use by adjacency:trans-uac (in)
```
Header Editor Example

This section contains the following examples:

- Example—Configuring and Applying the Header Editor, page 23-78
- Example—Using Directory Number Prefix to Set Privacy, page 23-79
- Example—Converting Remote-Party-ID or P-Preferred-Identity, page 23-80

Example—Configuring and Applying the Header Editor

The following example shows how to configure the EXAMPLE header editor and the abcd header type on the SBC2 SBC.

Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# sbc SBC2
Router(config-sbc)# sbe
Router(config-sbc-sbe)# sip header-editor EXAMPLE
Router(config-sbc-sbe-mep-hdr)# blacklist
Router(config-sbc-sbe-mep-hdr)# header abcd

The following example shows how the show sbc sbe sip header-editor command is used to display details of the EXAMPLE header editor:

Router# show sbc SBC2 sbe sip header-editor EXAMPLE
header-editor "EXAMPLE"
Description:
Type: Blacklist
store-rules:
No store-rule entries found.
request-line:
No request-line entries found.
headers:
abcd
entry 1
description:
action as-editor
Not in use with any adjacencies
Not in use with any method-editor

The header-editor inbound EXAMPLE command and the header-editor outbound EXAMPLE command applies the EXAMPLE header editor on the inbound and outbound direction. Therefore, for all incoming and outgoing messages, the header type abcd is checked. When the abcd header arrives or leaves, it is blacklisted and the error code 405 Method Not Allowed is generated. All the other headers are allowed.

Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# sbc SBC2
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency sip trans-uac
Router(config-sbc-sbe-adj-sip)# no attach
Router(config-sbc-sbe-adj-sip)# header-editor inbound EXAMPLE
Router(config-sbc-sbe-adj-sip)# header-editor outbound EXAMPLE
Router(config-sbc-sbe-adj-sip)# attach
The following example shows how the `show sbc sbe sip header-editor` command is used to display details of the EXAMPLE header editor after it has been applied to an adjacency.

Router# show sbc SBC2 sbe sip header-editor EXAMPLE

header-editor "EXAMPLE"
Description:
Type: Blacklist
store-rules:
No store-rule entries found.
request-line:
No request-line entries found.
headers:
abcd
entry 1
description:
   action as-editor
In use by adjacency: trans-uac (in, out)
Not in use with any method-editor

Example—Using Directory Number Prefix to Set Privacy

This example shows how to use a directory number prefix to set privacy:

Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# sip header-editor headprof1
Router(config-sbc-sbe-sip-hdr)# store-rule entry 1
Router(config-sbc-sbe-sip-hdr-ele-act)# description "store the called party number from To"
Router(config-sbc-sbe-sip-hdr-ele-act)# condition is-request eq true
Router(config-sbc-sbe-sip-hdr-ele-act)# condition and header-name To is-tel-uri eq true
Router(config-sbc-sbe-sip-hdr-ele-act)# condition and header-name To tel-uri-user store-as called-dn
Router(config-sbc-sbe-sip-hdr-ele-act)# exit
Router(config-sbc-sbe-sip-hdr)# store-rule entry 2
Router(config-sbc-sbe-sip-hdr-ele-act)# description "store the called party number from To"
Router(config-sbc-sbe-sip-hdr-ele-act)# condition is-request eq true

Router(config-sbc-sbe-sip-hdr-ele-act)# condition and header-name To is-sip-uri eq true
Router(config-sbc-sbe-sip-hdr-ele-act)# condition and header-name To sip-uri-user store-as called-dn
Router(config-sbc-sbe-sip-hdr-ele-act)# exit
Router(config-sbc-sbe-sip-hdr-ele-act)# store-rule entry 3
Router(config-sbc-sbe-sip-hdr-ele-act)# description "set $privacy based on DN"
Router(config-sbc-sbe-sip-hdr-ele-act)# condition variable privacy is-defined eq false
Router(config-sbc-sbe-sip-hdr-ele-act)# condition and variable called_dn is-defined eq true
Router(config-sbc-sbe-sip-hdr-ele-act)# condition and variable called_dn regex-match "^184"
Router(config-sbc-sbe-sip-hdr-ele-act)# condition and “none” store-as privacy
Router(config-sbc-sbe-sip-hdr-ele-act)# exit
Router(config-sbc-sbe-sip-hdr-ele-act)# store-rule entry 4
Router(config-sbc-sbe-sip-hdr-ele-act)# description "set $privacy based on DN"
Router(config-sbc-sbe-sip-hdr-ele-act)# condition variable privacy is-defined eq false
Router(config-sbc-sbe-sip-hdr-ele-act)# condition and variable called_dn is-defined eq true
Router(config-sbc-sbe-sip-hdr-ele-act)# condition and variable called_dn regex-match "^186"
Router(config-sbc-sbe-sip-hdr-ele-act)# condition and “user” store-as privacy
Example—Converting Remote-Party-ID or P-Preferred-Identity

This example converts Remote-Party-ID or From into P-Preferred-Identity. If the message is a request and Remote-Party-ID is present then it stores the username into a variable username. If the From header contains a sip: URI or Tel: URI, and Remote-Part-ID was not present then it stores the username into the variable username. Strips all P-Preferred-Identity, Remote-Party-ID’s and P-Preferred-Identity headers and inserts a single P-Preferred-Identity header containing the stored username and a Privacy header based on info received:

Router(config-sbc-sbe-sip-hdr-ele-act)# exit
Router(config-sbc-sbe-sip-hdr)# header Privacy entry 1
Router(config-sbc-sbe-sip-hdr-ele)# action strip
Router(config-sbc-sbe-sip-hdr-ele-act)# exit
Router(config-sbc-sbe-sip-hdr-ele)# header Privacy entry 2
Router(config-sbc-sbe-sip-hdr-ele)# action add-first-header value "($privacy)"
Router(config-sbc-sbe-sip-hdr-ele-act)# description "create a privacy header if we have privacy info"
Router(config-sbc-sbe-sip-hdr-ele-act)# condition variable privacy is-defined eq true

Associate with an inbound adjacency:

Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency sip adj1
Router(config-sbc-sbe-sip)# header-editor inbound headprof1

Example—Converting Remote-Party-ID or P-Preferred-Identity

This example converts Remote-Party-ID or From into P-Preferred-Identity. If the message is a request and Remote-Party-ID is present then it stores the username into a variable username. If the From header contains a sip: URI or Tel: URI, and Remote-Part-ID was not present then it stores the username into the variable username. Strips all P-Preferred-Identity, Remote-Party-ID’s and P-Preferred-Identity headers and inserts a single P-Preferred-Identity header containing the stored username and a Privacy header based on info received:

Router(config-sbc-sbe-sip-hdr-ele-act)# exit
Router(config-sbc-sbe-sip-hdr)# header Privacy entry 1
Router(config-sbc-sbe-sip-hdr-ele)# action strip
Router(config-sbc-sbe-sip-hdr-ele-act)# exit
Router(config-sbc-sbe-sip-hdr-ele)# header Privacy entry 2
Router(config-sbc-sbe-sip-hdr-ele)# action add-first-header value "($privacy)"
Router(config-sbc-sbe-sip-hdr-ele-act)# description "create a privacy header if we have privacy info"
Router(config-sbc-sbe-sip-hdr-ele-act)# condition variable privacy is-defined eq true

Associate with an inbound adjacency:

Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency sip adj1
Router(config-sbc-sbe-sip)# header-editor inbound headprof1

Example—Converting Remote-Party-ID or P-Preferred-Identity

This example converts Remote-Party-ID or From into P-Preferred-Identity. If the message is a request and Remote-Party-ID is present then it stores the username into a variable username. If the From header contains a sip: URI or Tel: URI, and Remote-Part-ID was not present then it stores the username into the variable username. Strips all P-Preferred-Identity, Remote-Party-ID’s and P-Preferred-Identity headers and inserts a single P-Preferred-Identity header containing the stored username and a Privacy header based on info received:

Router(config-sbc-sbe-sip-hdr-ele-act)# exit
Router(config-sbc-sbe-sip-hdr)# header Privacy entry 1
Router(config-sbc-sbe-sip-hdr-ele)# action strip
Router(config-sbc-sbe-sip-hdr-ele-act)# exit
Router(config-sbc-sbe-sip-hdr-ele)# header Privacy entry 2
Router(config-sbc-sbe-sip-hdr-ele)# action add-first-header value "($privacy)"
Router(config-sbc-sbe-sip-hdr-ele-act)# description "create a privacy header if we have privacy info"
Router(config-sbc-sbe-sip-hdr-ele-act)# condition variable privacy is-defined eq true

Associate with an inbound adjacency:

Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency sip adj1
Router(config-sbc-sbe-sip)# header-editor inbound headprof1

Example—Converting Remote-Party-ID or P-Preferred-Identity

This example converts Remote-Party-ID or From into P-Preferred-Identity. If the message is a request and Remote-Party-ID is present then it stores the username into a variable username. If the From header contains a sip: URI or Tel: URI, and Remote-Part-ID was not present then it stores the username into the variable username. Strips all P-Preferred-Identity, Remote-Party-ID’s and P-Preferred-Identity headers and inserts a single P-Preferred-Identity header containing the stored username and a Privacy header based on info received:

Router(config-sbc-sbe-sip-hdr-ele-act)# exit
Router(config-sbc-sbe-sip-hdr)# header Privacy entry 1
Router(config-sbc-sbe-sip-hdr-ele)# action strip
Router(config-sbc-sbe-sip-hdr-ele-act)# exit
Router(config-sbc-sbe-sip-hdr-ele)# header Privacy entry 2
Router(config-sbc-sbe-sip-hdr-ele)# action add-first-header value "($privacy)"
Router(config-sbc-sbe-sip-hdr-ele-act)# description "create a privacy header if we have privacy info"
Router(config-sbc-sbe-sip-hdr-ele-act)# condition variable privacy is-defined eq true

Associate with an inbound adjacency:

Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency sip adj1
Router(config-sbc-sbe-sip)# header-editor inbound headprof1

Example—Converting Remote-Party-ID or P-Preferred-Identity

This example converts Remote-Party-ID or From into P-Preferred-Identity. If the message is a request and Remote-Party-ID is present then it stores the username into a variable username. If the From header contains a sip: URI or Tel: URI, and Remote-Part-ID was not present then it stores the username into the variable username. Strips all P-Preferred-Identity, Remote-Party-ID’s and P-Preferred-Identity headers and inserts a single P-Preferred-Identity header containing the stored username and a Privacy header based on info received:

Router(config-sbc-sbe-sip-hdr-ele-act)# exit
Router(config-sbc-sbe-sip-hdr)# header Privacy entry 1
Router(config-sbc-sbe-sip-hdr-ele)# action strip
Router(config-sbc-sbe-sip-hdr-ele-act)# exit
Router(config-sbc-sbe-sip-hdr-ele)# header Privacy entry 2
Router(config-sbc-sbe-sip-hdr-ele)# action add-first-header value "($privacy)"
Router(config-sbc-sbe-sip-hdr-ele-act)# description "create a privacy header if we have privacy info"
Router(config-sbc-sbe-sip-hdr-ele-act)# condition variable privacy is-defined eq true

Associate with an inbound adjacency:

Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency sip adj1
Router(config-sbc-sbe-sip)# header-editor inbound headprof1

Example—Converting Remote-Party-ID or P-Preferred-Identity

This example converts Remote-Party-ID or From into P-Preferred-Identity. If the message is a request and Remote-Party-ID is present then it stores the username into a variable username. If the From header contains a sip: URI or Tel: URI, and Remote-Part-ID was not present then it stores the username into the variable username. Strips all P-Preferred-Identity, Remote-Party-ID’s and P-Preferred-Identity headers and inserts a single P-Preferred-Identity header containing the stored username and a Privacy header based on info received:

Router(config-sbc-sbe-sip-hdr-ele-act)# exit
Router(config-sbc-sbe-sip-hdr)# header Privacy entry 1
Router(config-sbc-sbe-sip-hdr-ele)# action strip
Router(config-sbc-sbe-sip-hdr-ele-act)# exit
Router(config-sbc-sbe-sip-hdr-ele)# header Privacy entry 2
Router(config-sbc-sbe-sip-hdr-ele)# action add-first-header value "($privacy)"
Router(config-sbc-sbe-sip-hdr-ele-act)# description "create a privacy header if we have privacy info"
Router(config-sbc-sbe-sip-hdr-ele-act)# condition variable privacy is-defined eq true

Associate with an inbound adjacency:

Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency sip adj1
Router(config-sbc-sbe-sip)# header-editor inbound headprof1
Router(config-sbc-sbe-sip-hdr-ele-act)# condition and header-name From header-uri
tel-uri-user store-as username
Router(config-sbc-sbe-sip-hdr-ele-act)# exit
Router(config-sbc-sbe-sip-hdr)# store-rule entry 5
test
Router(config-sbc-sbe-sip-hdr)# description "convert RPID param into Privacy header value"
Router(config-sbc-sbe-sip-hdr)# condition variable rpid_privacy is-defined eq true
Router(config-sbc-sbe-sip-hdr)# condition and variable rpid_privacy eq "off"
Router(config-sbc-sbe-sip-hdr)# condition and "none" store-as privacy
Router(config-sbc-sbe-sip-hdr-ele-act)# exit
Router(config-sbc-sbe-sip-hdr)# store-rule entry 6
test
Router(config-sbc-sbe-sip-hdr)# description "convert RPID param into Privacy header value"
Router(config-sbc-sbe-sip-hdr-ele-act)# condition variable rpid_privacy is-defined eq true
Router(config-sbc-sbe-sip-hdr-ele-act)# condition and variable rpid_privacy eq "id"
Router(config-sbc-sbe-sip-hdr-ele-act)# exit
Router(config-sbc-sbe-sip-hdr)# header P-Preferred-Identity entry 1
Router(config-sbc-sbe-sip-hdr-ele)# action strip
Router(config-sbc-sbe-sip-hdr-ele)# exit
Router(config-sbc-sbe-sip-hdr)# header P-Preferred-Identity entry 2
Router(config-sbc-sbe-sip-hdr-ele)# action add-first-header value "<sip:${username}@mydomain.com;user=phone>"
Router(config-sbc-sbe-sip-hdr-ele-act)# description "create a P-Preferred-Identity header"
Router(config-sbc-sbe-sip-hdr-ele-act)# condition variable username is-defined eq true
Router(config-sbc-sbe-sip-hdr-ele-act)# exit
Router(config-sbc-sbe-sip-hdr)# header P-Asserted-Identity entry 1
Router(config-sbc-sbe-sip-hdr-ele)# action strip
Router(config-sbc-sbe-sip-hdr-ele)# exit
Router(config-sbc-sbe-sip-hdr)# header Remote-Party-ID entry 1
Router(config-sbc-sbe-sip-hdr-ele)# action strip
Router(config-sbc-sbe-sip-hdr-ele)# exit
Router(config-sbc-sbe-sip-hdr)# header Privacy entry 1
Router(config-sbc-sbe-sip-hdr-ele)# action strip
Router(config-sbc-sbe-sip-hdr-ele)# exit
Router(config-sbc-sbe-sip-hdr)# header Privacy entry 2
Router(config-sbc-sbe-sip-hdr-ele)# action add-first-header value "${privacy}"
Router(config-sbc-sbe-sip-hdr-ele-act)# description "create a privacy header if we have privacy info"
Router(config-sbc-sbe-sip-hdr-ele-act)# condition variable privacy is-defined eq true

Associate with an inbound adjacency:

Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency sip adj1
Router(config-sbc-sbe-sip)# header-editor inbound headprof1

Body Editor Example

The following example shows how to configure the beditor1 body editor on the SBC2 SBC:

Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# sbc SBC2
Router(config-sbc)# sbe
Router(config-sbc-sbe)# sip body-editor beditor1
Chapter 23      SIP Message Manipulation

SIP Message Editing Using Editors

Router(config-sbc-sbe-mep-bdy)# body dtmf-relay/mixed
Router(config-sbc-sbe-mep-bdy-ele)# action reject

The following example shows how the show sbc sbe sip body-editor command is used to display details of the beditor1 body editor:

Router# show sbc SBC2 sbe sip body-editor beditor1

body-editor "beditor1"
  Description:
  Bodies:
    dtmf-relay/mixed
    action reject
    hunt-on-reject false
  Not in use with any adjacencies
  Not in use with any method-editor

Example—Applying Body Editor

The body-editor inbound beditor1 command and the body-editor outbound beditor1 command applies the beditor1 body editor on the inbound and outbound direction.

Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# sbc SBC2
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency sip trans-uac
Router(config-sbc-sbe-adj-sip)# no attach
Router(config-sbc-sbe-adj-sip)# body-editor inbound beditor1
Router(config-sbc-sbe-adj-sip)# body-editor outbound beditor1
Router(config-sbc-sbe-adj-sip)# attach

The following examples shows how the show sbc sbe sip body-editor command is used to display details of the beditor1 body editor after it has been applied to an adjacency:

Router# show sbc SBC2 sbe sip body-editor
body-editors for SBC service "SBC2"
 Name                        In use
-----------------------------
be1                          No
beditor1                     Yes
default                     No

Router# show sbc SBC2 sbe sip body-editor beditor1
body-editor "beditor1"
  Description:
  Bodies:
    dtmf-relay/mixed
    action reject
    hunt-on-reject false
  In use by adjacency:trans-uac (in, out)
  Not in use with any method-editor
Option Editor Example

The following example shows how to configure the oeditor1 option editor on the SBC2 SBC:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# sbc SBC2
Router(config-sbc)# sbe
Router(config-sbc-sbe)# sip option-editor oeditor1
Router(config-sbc-sbe)# option opt
```

Example—Applying Option Editor

The `option-editor inbound oeditor1` command and the `option-editor outbound oeditor1` command applies the oeditor1 option editor on the inbound and outbound direction.

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# sbc SBC2
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency sip trans-uac
Router(config-sbc-sbe-adj-sip)# no attach
Router(config-sbc-sbe-adj-sip)# option-editor ua inbound oeditor1
Router(config-sbc-sbe-adj-sip)# option-editor ua outbound oeditor1
Router(config-sbc-sbe-adj-sip)# attach
```

The following shows how the `show sbc sbe sip option-editor` command is used to display details of the oeditor1 option editor:

```
Router# show sbc SBC2 sbe sip option-editor oeditor1
option-editor "oeditor1"
   Description: Whitelist
   Options: opt
   In use by adjacency: ASR-15 (in-ua)
```

```
Router# show sbc SBC2 sbe sip option-editor
option editors for SBC service "SBC2"
Name In use
====================================
opt No
oeditor1 Yes
```

Parameter Editor Example

The following example shows how to configure the peditor1 parameter editor on the SBC2 SBC:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# sbc SBC2
Router(config-sbc)# sbe
Router(config-sbc-sbe)# sip parameter-editor peditor
Router(config-sbc-sbe-mep-prm)# parameter param
Router(config-sbc-sbe-mep-prm-ele)# action strip
```
Example—Applying Parameter Editor

The following example shows how to apply the peditor parameter editor to the he1 header editor on the SBC2 SBC:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# sbc SBC2
Router(config-sbc)# sbe
Router(config-sbc-sbe)# sip header-editor he1
Router(config-sbc-sbe-mep-hdr)# header Subject
Router(config-sbc-sbe-mep-hdr-ele)# parameter-editor peditor
```

The following shows how the `show sbc sbe sip header-editor` command is used to display details of the he1 header editor:

```
Router# show sbc SBC2 sbe sip header-editor he1
header-editor "he1"
  Description:
    Type:        Whitelist
    store-rules:
      No store-rule entries found.
    request-line:
      No request-line entries found.
    headers:
      subject
      entry 1
        description:
          action as-profile
          parameter-profile peditor
```

The following example shows how the `show sbc sbe sip parameter-editor` command is used to display details of the peditor parameter editor:

```
Router# show sbc SBC2 sbe sip parameter-editor peditor
parameter-editor "peditor"
  Description:
    Parameters:
      param
        action strip
    In use by header-editor:he1, header:subject, entry:1
```

SDP Editing Using Script-Based Editors

This section describes script-based editors for modifying the SDP content in SIP messages. The SIP Message Editing Using Editors section on page 23-64 describes body, header, method, option, and parameter editors that you directly configure on the SBC. You can apply any combination of script-based editors and directly configured editors to edit SIP messages.

From Release 3.4S, you can use scripts written using the Lua programming language to modify the SDP content in SIP messages. Typically, a Lua script consists of a group of one or more related functions. In the context of the SIP Message Editing feature, you write these functions with the objective of editing SIP messages. For detailed information about the Lua programming language, visit the Lua website at http://www.lua.org/.
You can configure a set of Lua scripts on the SBC. A set of scripts describes a set of editing actions to be applied to SIP messages. While configuring a script set, you specify the order in which scripts are loaded in the script set.

You can register the message-editing functions in the scripts as editors. These editors are called by the SBC at run time and applied to SIP messages. You can use these editors in conjunction with the body, header, method, option, and parameter editors configured on the SBC.

After you configure a script set, you can perform isolation testing and live testing on the script set to ensure that it works as expected.

At any point of time, only one script set can be active on the SBC. However, multiple script sets can be defined and kept ready for future use. You can switch a script set from the active state to the inactive state according to your requirements and vice versa.

The following sections provide information about creating Lua scripts and configuring script-based editing:

- Creating Lua Scripts for Script-Based Editing, page 23-85
- Configuring Script-Based Editors on the SBC, page 23-91
- Creating and Configuring Script-Based Editors: Examples, page 23-99

Creating Lua Scripts for Script-Based Editing

The following sections provide information that you can use while creating Lua scripts for script-based editing:

- Built-in Lua Classes, page 23-85
- Built-in Application Variables, page 23-89
- Built-in Logger Functions, page 23-90
- Built-in Register Function, page 23-90
- User-Defined Application Variables, page 23-91

Built-in Lua Classes

Lua scripts use an XPath-compatible method of referring to each node within the SDP body of a SIP message. The following example shows a sample SDP body in XML format. In the Lua code that you write, each XML tag can be uniquely identified by its path. A syntax-based function (such as the MeBlock:select_by_syntax function that is explained in the MeBlock Class section on page 23-86) can accept and process data based on the path that is passed to the function. A path is a forward-slash-separated string. For example, the sdpmedia[1]/line[3] path identifies the third line in the first media tag. Therefore, the sdpmedia[1]/line[3] path refers to b=AS:64.

```xml
<sdp>
  <line>v=0</line>
  <line>c=IN IP4 192.0.2.27</line>
  <line>s=-</line>
  <line>t=0 0</line>
  <line>r=604800 3600 0 90000</line>
  <line>a=foo</line>
  <media>
    <line>m=audio 32768 RTP/AVP 0 101</line>
    <line>c=IN IP4 0.0.0.0</line>
  </media>
</sdp>
```
You can use the following built-in Lua classes when writing scripts to modify the SDP body of SIP messages.

### MeMsg Class

An object of the MeMsg class contains the top-level structure of the message, which in turn, contains the entire SIP message. Table 23-4 describes the functions of this class.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>:get_sdp() or .sdp</td>
<td>Returns the MeBlock object that holds the SDP body.</td>
</tr>
<tr>
<td>:get_current_edit_point</td>
<td>Returns the current edit point, which is either before-receive or after-send.</td>
</tr>
<tr>
<td>:reject(error_code)</td>
<td>Fails a SIP request, or discards the response.</td>
</tr>
<tr>
<td>:get_app_variables() or .app_variable</td>
<td>Returns the table of application variables.</td>
</tr>
</tbody>
</table>

### MeBlock Class

An object of the MeBlock class represents a node in the SDP tree. A block is a contiguous subset of the SDP that is used for accessing strings. Table 23-5 describes the functions of the MeBlock class.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.new(syntax)</td>
<td>Constructs a block using the given syntax.</td>
</tr>
<tr>
<td>:get_type() or .type</td>
<td>Returns the syntax type (line, media, or sdp) of the MeBlock object.</td>
</tr>
<tr>
<td>:get_parent() or .parent</td>
<td>Returns the parent of this MeBlock object, which is either another MeBlock object or NIL for the root.</td>
</tr>
<tr>
<td>:get_children() or .children</td>
<td>Returns a MeSelection object that contains the child elements of the block.</td>
</tr>
<tr>
<td>:select_by_prefix(text_prefix)</td>
<td>Returns a MeSelection object containing all the lines of the MeBlock object that have the specified prefix.</td>
</tr>
</tbody>
</table>
MeSelection Class

An object of the MeSelection class is a list of MeBlock objects. Objects of the MeSelection class are used to process a set of lines. They can also be used to process child blocks in a parent block. A MeSelection object sequences MeBlock objects in the order in which they appear in the message. Table 23-6 describes the functions of the MeSelection class.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>:empty()</td>
<td>Returns TRUE if this selection is empty.</td>
</tr>
<tr>
<td>:iter()</td>
<td>Returns a generic For iterator that performs the specified action on all the objects in the MeSelection object. Each object is either of the MeBlock class or one of its subclasses.</td>
</tr>
<tr>
<td>[] operator</td>
<td>Use this one-based array operator to get the n-th block in the MeSelection object. Negative array indexes are also supported.</td>
</tr>
</tbody>
</table>

MeTextBlock Class

An object of the MeTextBlock class is used to assign, create, or manipulate text. Table 23-7 describes the functions of this class.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.new(type,text)</td>
<td>Constructs a new block of a specific type (line, media, and so on) using the specified text.</td>
</tr>
<tr>
<td>:get_text() or .text</td>
<td>Returns the text of the MeTextBlock object.</td>
</tr>
</tbody>
</table>
Table 23-7  Functions of the MeTextBlock Class (continued)

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>:set_text()</td>
<td>Sets the text of the MeTextBlock object. Note that existing text is replaced when this function is called.</td>
</tr>
<tr>
<td>:find(args)</td>
<td>Calls string.find(args) on the text of this MeTextBlock object.</td>
</tr>
<tr>
<td>:match(args)</td>
<td>Calls string.match(args) on the text of this MeTextBlock object.</td>
</tr>
<tr>
<td>:replace(args)</td>
<td>Calls string.gsub(args) on the text of this MeTextBlock object.</td>
</tr>
</tbody>
</table>

MeSdp Class

An object of the MeSdp class is used to retrieve specific parts of the SDP body. Table 23-8 describes the functions of this class.

Table 23-8  Functions of the MeSdp Class

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>:get_session_lines() or .session_lines</td>
<td>Returns a MeSelection object containing the session lines of the SDP body.</td>
</tr>
<tr>
<td>:get_media_blocks() or .media_blocks</td>
<td>Returns a MeSelection object containing the media blocks.</td>
</tr>
</tbody>
</table>

MeSdpMedia Class

An object of the MeSdpMedia class is used to create or retrieve SDP media blocks. Table 23-9 describes the functions of this class.

Table 23-9  Functions of the MeSdpMedia Class

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.new(text)</td>
<td>Constructs a block of the media syntax (or block type) using the specified text.</td>
</tr>
<tr>
<td>:get_media_lines() or .media_lines</td>
<td>Returns a MeSelection object containing the media lines of the MeSdpMedia object.</td>
</tr>
</tbody>
</table>
MeSdpLine Class

An object of the MeSDPLine class is used to create a line in the SDP message. Table 23-10 describes the functions of this class.

Table 23-10  Functions of the MeSdpLine Class

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.new(text)</td>
<td>Constructs a block of the line syntax (or block type) using the specified text.</td>
</tr>
</tbody>
</table>

Built-in Application Variables

This section describes the built-in application variables that you can use while writing Lua scripts. Built-in application variables, such as configuration data for an adjacency and transport values, are initialized by the SBC and are available to the Lua scripts. They are read-only, start with the characters msg. or adj., and are reserved because you cannot create variables with these prefixes. Table 23-11 describes the built-in application variables that can be accessed within a script.

Table 23-11  Built-in Application Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>adj.account</td>
<td>Adjacency account.</td>
</tr>
<tr>
<td>adj.dest_addr</td>
<td>Adjacency signaling peer.</td>
</tr>
<tr>
<td>adj.dest_port</td>
<td>Adjacency signaling peer port.</td>
</tr>
<tr>
<td>adj.group</td>
<td>Adjacency group.</td>
</tr>
<tr>
<td>adj.home_net_id</td>
<td>Adjacency home network identity.</td>
</tr>
<tr>
<td>adj.ip_realm</td>
<td>Adjacency realm.</td>
</tr>
<tr>
<td>adj.lcl_addr</td>
<td>Adjacency signaling address.</td>
</tr>
<tr>
<td>adj.lcl_port</td>
<td>Adjacency signaling port.</td>
</tr>
<tr>
<td>adj.lcl_id</td>
<td>Adjacency local ID.</td>
</tr>
<tr>
<td>adj.listen_trans</td>
<td>Adjacency listen transport.</td>
</tr>
<tr>
<td>adj.mandatory_trans</td>
<td>Adjacency mandatory transport.</td>
</tr>
<tr>
<td>adj.med_loc</td>
<td>Adjacency media location.</td>
</tr>
<tr>
<td>adj.name</td>
<td>Adjacency name.</td>
</tr>
<tr>
<td>adj.preferred_trans</td>
<td>Adjacency preferred transport.</td>
</tr>
<tr>
<td>adj.trust_level</td>
<td>Adjacency trust level.</td>
</tr>
<tr>
<td>adj.target_reg_addr</td>
<td>Adjacency registration target address.</td>
</tr>
<tr>
<td>adj.targrt_reg_port</td>
<td>Adjacency registration target port.</td>
</tr>
<tr>
<td>adj.visited_net_id</td>
<td>Adjacency visited network identity.</td>
</tr>
<tr>
<td>adj.vpn_id</td>
<td>Adjacency VPN ID.</td>
</tr>
</tbody>
</table>
Chapter 23      SIP Message Manipulation

SDP Editing Using Script-Based Editors

**Table 23-11   Built-in Application Variables**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>msg.status_code</td>
<td>Response status code. Returns the string representation of the status code for a SIP response message. Returns an empty string for a SIP request message.</td>
</tr>
<tr>
<td>msg.header(&quot;name&quot;).value</td>
<td>Value of the first header with the name name in the message. Only nonvital headers can be used with this function.</td>
</tr>
<tr>
<td>msg.header(&quot;name&quot;).uri.tel_uri.number</td>
<td>Directory number of the TEL URI in the first header with the name name. If used on a SIP URI, an empty string is returned. Only To and From headers can be used with this function.</td>
</tr>
<tr>
<td>msg.header(&quot;name&quot;).uri.sip_uri.user</td>
<td>User name of the SIP URI or SIPS URI in the first header with the name name. If used on a TEL URI, an empty string is returned. Only To and From headers can be used with this function.</td>
</tr>
<tr>
<td>msg.lcl_ip_addr</td>
<td>Address at which the message was received.</td>
</tr>
<tr>
<td>msg.lcl_port</td>
<td>Port at which the message was received.</td>
</tr>
<tr>
<td>msg.rmt_ip_addr</td>
<td>Previous hop IP address.</td>
</tr>
<tr>
<td>msg.rmt_port</td>
<td>Previous hop port.</td>
</tr>
</tbody>
</table>

**Built-in Logger Functions**

The following logger functions can be called to create logs:

- MeLogger.debug(text) Log at debug level (30)
- MeLogger.detail(text) Log at detail level (50)
- MeLogger.info(text) Log at info level (60)
- MeLogger.config(text) Log at config level (63)
- MeLogger.warn(text) Log at warn level (70)
- MeLogger.error(text) Log at error level (80)

**Built-in Register Function**

Use the following function to register functions as editors with the SBC:

MeEditor.register(edit_point,editor_name,edit_func)

By including this line in the script, you can register a function as an editor with the SBC, assign the function a name as an editor, and specify the point at which the function must be applied as an editor on SIP messages.

The following are the arguments of the MeEditor.register function:

- edit_point—Accepts one of the following values:
  - AFTER_SEND—Specifies that the outgoing message must be edited after it is processed by the adjacency and just before it is forwarded from the adjacency.
– **BEFORE_RECEIVE**—Specifies that the incoming message must be edited just after it is received on the adjacency and before the adjacency begins processing it.

- **editor_name**—Specifies the name that you want to assign to the editor.

**Note**: Names that you assign to editors in a script set must be unique. However, editors in different script sets can have the same name.

- **edit_func** is the name of the function in the script that you want to designate as an editor.

The following example shows how to register the `hello_world` Lua function as an editor:

```plaintext```
MeEditor.register(MeEditor.BEFORE_RECEIVE, 
                 "hello_world_editor", 
                 hello_world)
```

### User-Defined Application Variables

User-defined application variables are used to pass user information among Lua edit functions and between script-based editors and editors that are directly configured on the SBC that is, body, header, method, option, and parameter editors.

### Configuring Script-Based Editors on the SBC

This task shows how to configure a script-based editor on the SBC.

**Note**: Before you start performing this task, create the scripts offline and place the script files at a location from where they can be accessed from the SBC. Copy the script files to the SBC by using trivial file transfer protocol (TFTP), file transfer protocol (FTP), remote copy protocol (rcp), secure copy protocol (SCP), or any other supported application.

### SUMMARY STEPS

1. configure terminal
2. sbc sbc-name
3. sbe
4. script-set set-number lua
5. script script-name
6. load-order order-index-number
7. type {full | wrapped edit-point {before-receive | after-send | both}}
8. filename {bootflash: | flash: | fpd: | nvram: | obfl: | any-other-device}
9. exit
10. complete
11. end
12. test sbc message sip filename device-type:file-name script-set script-set-number {after-send | before-receive} editors {editor1-name [editor2-name] [editor3-name] . . . [editor8-name]}
13. configure terminal
14. sbc sbc-name
15. sbe
16. adjacency sip adjacency-name
17. test script-set set-number
18. exit
19. active-script-set script-set-number
20. adjacency sip adjacency-name
21. editor-list { after-send | before-receive }
22. editor order-number editor-name [ condition [ body contains sdp ] ]
23. end
24. show sbc sbc-name sbe script-set set-number [ script script-name [ line-numbers ] | program [ line-numbers ] | statistics ]
25. clear sbc sbc-name sbe script-set-stats set-number [ editors-stats editor-name ]
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters the global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc sbc-name</td>
<td>Enters the SBC service mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# sbc mysbc</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Enters the SBE configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> script-set set-number lua</td>
<td>Configures a script set composed of scripts written using the Lua programming language.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe)# script-set 20 lua</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> script script-name</td>
<td>Configures a script in the script set. Note that multiple scripts can be configured in a script set.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-script-set)# script SBCScript</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> load-order order-index-number</td>
<td>Specifies the load order of the script. Scripts are loaded in ascending order of the order index number. For example, a script with the order index number 4 is loaded before a script with the order index number 6.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-scrpset-script)# load-order 2</td>
<td></td>
</tr>
</tbody>
</table>

- **sbc-name**—Name of the SBC.

- **set-number**—Script set number. The range is from 1 to 2147483647.

- **script-name**—Name of the script.

- **order-index-number**—Load order index number. The range is from 1 to 4294967295. The default order index number is 100. For scripts that are subsequently added and for which the **load-order** command is not run, the default order index number is set in multiples of 100 (that is, 200, 300, 400, and so on).
# Chapter 23      SIP Message Manipulation

## SDP Editing Using Script-Based Editors

### Command or Action

| Step 7 | type {full | wrapped edit-point (after-send | before-receive | both)) |
|--------|--------------------------------------|
| Example: | Router(config-sbc-sbe-scrpset-script)# type full |

Specifies the script type:
- **full**—Specifies a full script without autogeneration.
- **wrapped edit-point**—Specifies a script that must be autogenerated from the file and the edit point to be used in autoregistration. One of the following edit points can be specified:
  - **after-send**—Specifies that the outgoing message must be edited after the message is processed by the adjacency and just before it is forwarded from the adjacency.
  - **before-receive**—Specifies that the incoming message must be edited just after it is received on the adjacency and before the adjacency begins processing it.
  - **both**—Enables editing of the message both before and after it is processed by the SBC.

<table>
<thead>
<tr>
<th>Step 8</th>
<th>filename {device-type:file-path-and-name}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-scrpset-script)# filename bootflash:lua1.lua</td>
</tr>
</tbody>
</table>

Specifies the path and name of the script file.
- **device-type**—One of the following or any other storage device installed on the router:
  - **bootflash:**
  - **flash:**
  - **fpd:**
  - **nvram:**
  - **obfl:**
  The list of file system devices is dynamically generated and displayed. Other devices, such as a hard disk, that are available on the platform can also be used in this command.
- **file-path-and-name**—Full path and name of the script file on the specified storage device.

<table>
<thead>
<tr>
<th>Step 9</th>
<th>exit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-scrpset-script)# exit</td>
</tr>
</tbody>
</table>

Exits the script configuration mode and enters the script-set configuration mode.

<table>
<thead>
<tr>
<th>Step 10</th>
<th>complete</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-script-set)# complete</td>
</tr>
</tbody>
</table>

Validates and loads the scripts. If syntax errors are encountered during the validation process, error messages are displayed. If a script is syntactically correct, it is loaded into memory and the editors are registered with the Lua run-time environment. If required, you can switch to the privileged EXEC mode and then run the **show sbc sbe editors** command to verify that the editors are correctly registered.
### Command or Action

<table>
<thead>
<tr>
<th>Step 11</th>
<th>end</th>
</tr>
</thead>
</table>

**Example:**
Router(config-sbc-sbe-script-set)# end

Exits the script-set configuration mode, and returns to the privileged EXEC mode.

| Step 12 | test sbc message sip filename device-type:file-name script-set script-set-number (after-send | before-receive) editors {editor1-name [editor2-name] [editor3-name] ... [editor8-name]}

**Example:**
Router# test sbc message sip filename bootflash:inv script-set 123 after-send editors sdp_add_after my-header-editor

Performs isolation testing of script-based editors.

**Note**
Although it is optional to perform isolation testing, we recommend that you perform the procedure. See the **Isolation Testing of Script-Based Editors: Example?** section on page 23-100 for detailed information about the procedure.

- **device-type**—Type of storage device on which you have stored the file containing the SIP message on which you want to test the editors. In the command-line interface (CLI), when you enter a question mark after the **test sbc message sip filename script-set editors** command, a list of all the storage devices installed on the router is displayed. The device can be one of the following or any other storage device installed on the router:
  - bootflash:
  - flash:
  - fpd:
  - nvram:
  - obfl:

  The list of file system devices is dynamically generated and displayed. Other devices, such as a hard disk, that are available on the platform can also be used in this command.

- **file-name**—Name of the file containing the SIP message on which you want to test the editors.

- **script-set-number**—Number of the script set containing the editors that you want to test.

- **after-send**—Specifies that the outgoing message must be edited after the message is processed by the adjacency and just before it is forwarded from the adjacency.

- **before-receive**—Specifies that the incoming message must be edited just after it is received on the adjacency and before the adjacency begins processing it.

- **editor1-name . . . editor8-name**—Names of the editors. You can specify up to eight editors. You must specify at least one editor.
### SDP Editing Using Script-Based Editors

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>configure terminal</td>
<td>Enters the global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>sbc sbc-name</td>
<td>Enters the SBC service mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config)# sbc mysbc</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>sbe</td>
<td>Enters the SBE configuration mode of the SBC.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>adjacency sip adjacency-name</td>
<td>Enters the SBE SIP adjacency configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe)# adjacency sip adj1</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>test script-set script-set-number</td>
<td>Performs live testing of script-based editors.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-adj-sip)# test script-set 123</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>exit</td>
<td>Exits the SIP adjacency configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-adj-sip)# exit</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>active-script-set script-set-number</td>
<td>Activates the script set.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe)# active-script-set 20</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>adjacency sip adjacency-name</td>
<td>Enters the SBE SIP adjacency configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe)# adjacency sip adj1</td>
<td></td>
</tr>
</tbody>
</table>

**Note** Although it is optional to perform live testing, we recommend that you perform the procedure. See the Live Testing of Script-Based Editors: Example section on page 23-102 for detailed information.

**Command or Action**

- **configure terminal**
  - Enters the global configuration mode.

**Purpose**

- **sbc sbc-name**
  - Enters the SBC service mode.
  - *sbc-name*—Name of the SBC.

**Example:**

- Router# configure terminal

**Step 14**

**Example:**

- Router(config)# sbc mysbc

**Step 15**

**Example:**

- Router(config-sbc)# sbe

**Step 16**

**Example:**

- Router(config-sbc-sbe)# adjacency sip adj1

**Step 17**

**Example:**

- Router(config-sbc-sbe-adj-sip)# test script-set 123

**Step 18**

**Example:**

- Router(config-sbc-sbe-adj-sip)# exit

**Step 19**

**Example:**

- Router(config-sbc-sbe)# active-script-set 20

**Step 20**

**Example:**

- Router(config-sbc-sbe)# adjacency sip adj1
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 21</strong></td>
<td>**editor-list (after-send</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-adj-sip)# editor-list after-send</td>
</tr>
<tr>
<td><strong>Step 22</strong></td>
<td><strong>editor order-number editor-name [condition [body contains sdp]]</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-adj-sip-ed)# editor 2 sdp_add_after condition body contains sdp</td>
</tr>
<tr>
<td><strong>Step 23</strong></td>
<td><strong>end</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-adj-sip)# end</td>
</tr>
</tbody>
</table>

**Command or Action**

**Step 21**

`editor-list (after-send | before-receive)`

**Example:**

Router(config-sbc-sbe-adj-sip)# editor-list after-send

**Step 22**

`editor order-number editor-name [condition [body contains sdp]]`

**Example:**

Router(config-sbc-sbe-adj-sip-ed)# editor 2 sdp_add_after condition body contains sdp

**Step 23**

`end`

**Example:**

Router(config-sbc-sbe-adj-sip)# end

**Purpose**

**Step 21**

**editor-list (after-send | before-receive)**

Configures a list of editors.

- **after-send**—Specifies that the outgoing message must be edited after the message is processed by the adjacency and just before it is forwarded from the adjacency.
- **before-receive**—Specifies that the incoming message must be edited just after it is received on the adjacency and before the adjacency begins processing it.

**Step 22**

**editor order-number editor-name [condition [body contains sdp]]**

Configures an editor in the editor list. For each editor that you want to apply in a sequence, run this command to specify the order of the editor in the editor list.

**Note**

You can add any combination of script-based editors and body, header, method, option, and parameter editors in the editor list.

- **order-number**—Order in which the editor must be applied. The range is from 1 to 2147483647.
- **editor-name**—Name of the editor that you want to apply to messages that are processed by the adjacency.
- **condition**—Specifies that there are one or more conditions for the editor to be applied.
- **body contains sdp**—Specifies that the message body must be SDP-based content. The editor is applied only if this condition is met. Include `body contains sdp` in the command for script-based editors.

**Step 23**

**end**

Exits the SIP editor configuration mode, and enters the privileged EXEC mode.
### Command or Action

**Step 24**

```
show sbc sbc-name sbe script-set set-number [script script-name [line-numbers] | program [line-numbers] | statistics]
```

**Example:**
```
Router# show sbc mySbc sbe script-set 20 script SBCScript line-numbers
```

**Purpose**
Displays a summary of the details pertaining to all the configured script sets or shows the details of the specified script set.
- `sbc-name`—Name of the SBC.
- `set-number`—Script set number. The range is from 1 to 2147483647.
- `program`—Specifies that all scripts must be displayed as a single program.
- `line-numbers`—Specifies that line numbers must be included while displaying scripts.
- `script`—Specifies that details of a single script from the script set must be displayed.
- `script-name`—Name of the script that must be displayed.
- `statistics`—Specifies that script set statistics must be displayed.

**Step 25**

```
clear sbc sbc-name sbe script-set-stats script-set-number [editors-stats editor-name]
```

**Example:**
```
Router# clear sbc mySbc sbe script-set-stats 1
```

**Purpose**
Clears previously stored statistics related to a script set.
- `sbc-name`—Name of the SBC.
- `script-set-number`—Script set number. The range is from 1 to 2147483647.
- `editor-stats`—Specifies that the script set statistics must be cleared for a specific editor.
- `editor-name`—Name of the editor.

---

The following example shows how the `show sbc sbe script-set` command is used to display the summary of a script set:
```
Router# show sbc mySbc sbe script-set 1
name    language complete active status
--------- ----------- --------- ------- -------
script-set 1  lua yes      no   ok
script
--------- ----------- --------- ------- -------
edit_invite_1 1  bootflash:lua_1.lua
edit_invite_2 2  bootflash:lua_2.lua
edit_invite_3 3  bootflash:lua_3.lua
```
Creating and Configuring Script-Based Editors: Examples

The following sections describe how to create and configure sample script-based editors:

- Creating Lua Scripts: Example, page 23-99
- Configuring Script-Based Editors: Example, page 23-100
- Isolation Testing of Script-Based Editors: Example, page 23-100
- Live Testing of Script-Based Editors: Example, page 23-102

Creating Lua Scripts: Example

The following sections provide listings of sample Lua scripts:

- Adding Text in the SDP Body: Example, page 23-99
- Deleting Lines from the SDP Body: Example, page 23-100
- Replacing Text in the SDP Body: Example, page 23-100

Adding Text in the SDP Body: Example

The following example shows a Lua script that is used to add \texttt{sdp\_add\_after added this line} at the end of the SDP body:

\begin{verbatim}
function append_text(msg)
    msg.sdp:insert_child_last(MeSdpLine.new("sdp_add_after added this line"))
end
\end{verbatim}

The following example shows the line that registers the append\_text Lua function from the preceding example as an editor. In this example, the editor is named \texttt{sdp\_add\_after}.

\begin{verbatim}
MeEditor.register(MeEditor.BEFORE_RECEIVE,"sdp_add_after",append_text)
\end{verbatim}

\textbf{Note}

An editor is registered with the SBC when the script set containing the script with the editor code is configured on the SBC.

The complete code listing for this script is as follows:

\begin{verbatim}
function append_text(msg)
    msg.sdp:insert_child_last(MeSdpLine.new("sdp_add_after added this line"))
end
MeEditor.register(MeEditor.BEFORE_RECEIVE,"sdp_add_after",append_text)
\end{verbatim}

You can save these lines in a .lua file, copy the file to the SBC, and then perform the procedure described in the ?$paranum>Configuring Script-Based Editors on the SBC? section on page 23-91 to configure and test the editor.
Deleting Lines from the SDP Body: Example

The following script deletes all the lines that start with `a=deleteme` from the SDP bodies of SIP messages:

```lua
function delete_lines(msg)
    for line in msg.sdp:select_by_prefix("a=deleteme"):iter() do
        line:delete()
    end
end
MeEditor.register(MeEditor.BEFORE_RECEIVE,"Delete_a_Lines",delete_lines)
```

Replacing Text in the SDP Body: Example

The following script replaces `rtpmap` in the SDP body with `srtp_remap`:

```lua
function replace_text(msg)
    msg.sdp:replace("rtpmap","srtp_remap")
end
MeEditor.register(MeEditor.AFTER_SEND,"Switch_Protocol",replace_text)
```

Configuring Script-Based Editors: Example

The following example shows how to configure the script set created in the preceding example:

```lua
Router# configure terminal
Router(config)# sbc mysbc
Router(config-sbc)# sbe
Router(config-sbc-sbe)# script-set 20 lua
Router(config-sbc-sbe-script-set)# script SBCScript
Router(config-sbc-sbe-scrpset-script)# type full
Router(config-sbc-sbe-scrpset-script)# filename bootflash:lua1.lua
Router(config-sbc-sbe-scrpset-script)# exit
Router(config-sbc-sbe-script-set)# complete
Router(config-sbc-sbe-script-set)# end
Router# test sbc message sip filename bootflash:inv script-set 123 after-send editors sdp_add_after my-header-editor
Router# configure terminal
Router(config)# sbc mysbc
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency sip adj1
Router(config-sbc-sbe-adj-sip)# test script-set 123
Router(config-sbc-sbe-adj-sip)# exit
Router(config-sbc-sbe)# active-script-set 20
Router(config-sbc-sbe)# adjacency sip adj1
Router(config-sbc-sbe-adj-sip)# editor-list after-send
Router(config-sbc-sbe-adj-sip)# editor 2 sdp_add_after condition body contains sdp
Router(config-sbc-sbe-adj-sip)# end
Router# show sbc mysbc sbe script-set 20 script SBCscript line-numbers
```

Isolation Testing of Script-Based Editors: Example

During isolation testing of script-based editors, the SDP editing configuration is tested in isolation. No other form of SBC processing takes place. Isolation testing does not show interactions between the editing configuration and other configurations, such as, number validation configuration.

The `test sbc message` command is used to perform isolation testing on SIP messages. This command loads a file containing a valid protocol message and applies a list of user-specified editors to the message. It does not display details of interactions between editing and routing decisions. Up to eight editors can
be specified in the command. The order in which the editors are specified is the order in which they are applied. Note that profile editors that are not part of any specific script set can also be specified in the command.

In the following example, sdp_add_after is defined in script-set 123 and my_header_editor has been configured using the \texttt{sip header-editor} command. The sdp_add_after editor is the one used in the preceding sections describing examples. The lines highlighted in bold show the actions performed by the editors.

\begin{verbatim}
Router# test sbc message sip filename bootflash:inv script-set 123 after-send editors sdp_add_after my-header-editor

INVITE sip:john@example.com:55060 SIP/2.0
Via: SIP/2.0/UDP 192.0.2.195;branch=z9hG4bKff9b46fb055c0521cc24024da96cd290
Via: SIP/2.0/UDP 192.0.2.195:55061;branch=z9hG4bK291d90e31a47b225bd0ddf4353e9c0
From: <sip:192.0.2.195:55061;user=phone>;tag=GR52RWG346-34
To: "john@example.com" <sip:john@example.com:55060>
Call-ID: 12013223@192.0.2.195
CSeq: 1 INVITE
Contact: <sip:192.0.2.195:55060>
Content-Type: application/sdp
Content-Length: 229

v=0
o=Clarent 120386 120387 IN IP4 192.0.2.196
s=Clarent CSCM
c=IN IP4 192.0.2.196
t=0 0
m=audio 40376 RTP/AVP 8 18 4 0
a=rtpmap:8 PCMA/8000
a=rtpmap:18 G729/8000
a=rtpmap:4 G723/8000
a=rtpmap:0 PCMU/8000
a=SendRecv

%Test successful, edited message:
INVITE sip:john@example.com:55060 SIP/2.0
Via: SIP/2.0/UDP 192.0.2.195;branch=z9hG4bKff9b46fb055c0521cc24024da96cd290
Via: SIP/2.0/UDP 192.0.2.195:55061;branch=z9hG4bK291d90e31a47b225bd0ddf4353e9c0
From: <sip:192.0.2.195:55061;user=phone>;tag=GR52RWG346-34
To: "john@example.com" <sip:john@example.com:55060>
Call-ID: 12013223@192.0.2.195
CSeq: 1 INVITE
Contact: <sip:192.0.2.195:55060>
Content-Type: application/sdp
Content-Length: 258
name: cisco

v=0
o=Clarent 120386 120387 IN IP4 192.0.2.196
s=Clarent CSCM
c=IN IP4 192.0.2.196
t=0 0
m=audio 40376 RTP/AVP 8 18 4 0
a=rtpmap:8 PCMA/8000
a=rtpmap:18 G729/8000
a=rtpmap:4 G723/8000
a=rtpmap:0 PCMU/8000
a=SendRecv

sdp_add_after added this line
\end{verbatim}
Live Testing of Script-Based Editors: Example

During live testing of script-based editors, an adjacency is configured as a test adjacency. Inbound editing and outbound editing of messages on that adjacency are then performed using the script set specified in the test script-set command instead of the script set that is currently active. The following is a sample command:

Router(config-sbc-sbe-adj-sip)# test script-set 123

Note

The active script set is specified by the active-script-set command. You must ensure that the active-script-set command has not been run on the script set on which you run the test script-set command.

The test script-set command cannot be used to verify profile editors because the profile editors are not associated with a script set. To include a profile editor in the test, first configure the profile editor on the test adjacency by using the editor-list command.
Signaling Congestion Handling

Cisco Unified Border Element (SP Edition) supports signaling congestion handling to improve performance when external events can cause large bursts of user activity that exceed the capacity of the SBC. Previously, SBC discarded packets in these situations causing the sending endpoint to retransmit the packet, which increases the load on the system, increasing the latency or drop-rate further. Only a small proportion of calls succeed (much less than the rated capacity) and take a significant length of time to connect.

With congestion handling enhancements, SBC improves the successful call setup and registration rate under loads up to at least double its rated capacity. This is done by rejecting SIP calls or REGISTERs that cannot be processed to prevent retransmissions. The reject message contains a random RETRY-AFTER header that informs the sending endpoint when to send a retry.

The main advantages of rejecting incoming work are to:

- prevent retransmissions
- keep the latency of the system at an acceptable level

---

**Note**

This feature is supported in the unified model for Cisco IOS XE Release 2.5 and later.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).

**Feature History for Signaling Congestion Handling**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 2.5</td>
<td>This feature was introduced on the Cisco IOS XR.</td>
</tr>
</tbody>
</table>

---

**Contents**

This module contains the following sections:

- Restrictions for Signaling Congestion Handling, page 24-2
- Configuring Signaling Congestion Handling, page 24-2
Restrictions for Signaling Congestion Handling

The following restrictions apply when you configure the congestion handling enhancements on the Cisco Unified Border Element (SP Edition):

- SBC supports signaling congestion handling only at the global SBC congestion level. Congestion handling does not improve flow control from particular work-sources or sinks.
- Signaling congestion handling addresses SIP signaling workloads from out-of-dialog requests and only tests INVITE and REGISTER messages.

Configuring Signaling Congestion Handling

Signaling congestion handling is turned on by default; nevertheless it could be configured to change the reject message code.

Note

The reject message code is the code sent back to sender during congestion. Default reject message code is 503.

Cisco Unified Border Element (SP Edition) requires following configurations to enable signaling congestion handling enhancements:

SUMMARY STEPS

1. configure terminal
2. sbc sbc-name
3. sbe
4. congestion sip reject-code valid-reject-code
5. exit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# config terminal</td>
<td></td>
</tr>
<tr>
<td>Step 2 sbc sbc-name</td>
<td>Enters session border controller (SBC) configuration submode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config)# sbc test</td>
<td></td>
</tr>
<tr>
<td>Step 3 sbe</td>
<td>Enters signaling border element (SBE) configuration submode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc)# sbe</td>
<td></td>
</tr>
</tbody>
</table>
### Step 4

**congestion sip reject-code**

valid-reject-code

#### Example:

```bash
Router(config-sbc-sbe)# congestion sip reject-code 350
```

Changes the reject message code for congestion handling. The default reject message code is 503.

### Step 5

**exit**

#### Example:

```bash
Router(config-sbc-sbe)# exit
```
SIP IP-FQDN URI Translation

Cisco Unified Border Element (SP Edition) supports translation between IP addresses and fully-qualified domain names (FQDNs) in the Request-URI, To header, and From header in Session Initiation Protocol (SIP) messages, permitting SBC to interconnect SIP devices, expecting specific SIP URI formats.

Note
This feature is supported in the unified model for Cisco IOS XE Release 2.5 and later.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).

Feature History for IP to FQDN URI Translation Support

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 2.5</td>
<td>This feature was introduced on the Cisco IOS XR.</td>
</tr>
</tbody>
</table>

Contents

This module contains the following sections:

- Restrictions for SIP IP-FQDN URI Translation, page 25-1
- Information About SIP IP-FQDN URI Translation, page 25-2
- Configuring SIP IP-FQDN URI Translation, page 25-3
- Configuration Example for SIP IP-FQDN URI Translation, page 25-4

Restrictions for SIP IP-FQDN URI Translation

The following restrictions apply when you configure the SIP IP-FQDN URI translation on the Cisco Unified Border Element (SP Edition):

- Each IP address and FQDN required for translation must be explicitly configured on Cisco Unified Border Element (SP Edition).
- Cisco Unified Border Element (SP Edition) configures only one mapping for each IP address and FQDN.
• The FQDN must be unique when a bidirectional mapping is configured. SBC can configure multiple IP addresses that map to a single FQDN (IP1 -> FQDN1; IP2 -> FQDN1), but it cannot configure a single FQDN to map to multiple IP addresses.

## Information About SIP IP-FQDN URI Translation

### URI Translation

The URIs in the Request-URI, To header, and From header are translated based on the configuration of the ingress and egress adjacency.

The domain part of the SIP or SIPs URI is converted after the translation is configured and a mapping is found in the IP-to-FQDN mapping table. If no mapping is found, the SIP request is forwarded without any modification in the domain part.

For example, the following INVITE (with only the relevant parts shown):

```plaintext
INVITE sip:conf-server@12.34.56.78 SIP/2.0
From: End-User <sip:end-user@100.101.102.103>;tag=5678-EFGH
To: Conf-Server <sip:conf-server@12.34.56.78>
```

can be converted to:

```plaintext
INVITE sip:conf-server@example1.com SIP/2.0
From: End-User <sip:end-user@example2.com>;tag=5678-EFGH
To: Conf-Server <sip:conf-server@example1.com>
```

The reverse conversion is also possible.

The To and From headers are established at initial call setup and are not changed later. Hence, any change in the mapping table does not affect existing calls.

For outbound requests on an adjacency, translation can be configured in one direction only, such as from IP address to FQDN. The domain part of the URI is converted only if it is an IP address. If it is already an FQDN, no conversion is required. Similarly, for inbound requests on an adjacency, translation can be configured in one direction only; this would typically be the other direction from inbound requests.

To prevent DNS lookups for outgoing Request-URI, IP->FQDN translations configure “force-signaling-peer”, which enforces strict routing.

An exact case-insensitive match is required to translate an FQDN. For example, the mapping "aaa.com <-> 1.1.1.1" would not match the domain "web.aaa.com", but it would match the domain “AAA.com”.

**Note** If SIP to Tel URI conversion is also configured for the adjacency, this takes precedence over translation between IP address and FQDN. If the egress adjacency is configured to rewrite the To header or From header, this also takes precedence over translation between IP address and FQDN.
Configuring SIP IP-FQDN URI Translation

Cisco Unified Border Element (SP Edition) provides a configurable mapping between IP addresses and FQDNs to convert an IP address to a FQDN and a FQDN to an IP address in the Request-URI, To header, and From header. This behavior is configurable on a SIP adjacency, which can be either or both of the ingress and egress adjacency. When configured on the ingress adjacency, the translated URI is used as an input to routing, screening, data modification, CAC policy, and CDRs.

This section contains the steps to configure the SIP IP-FQDN URI Translation feature.

SUMMARY STEPS

1. configure terminal
2. sbc sbc-name
3. sbe
4. adjacency sip adjacency-name
5. translate {request-uri | to | from} {inbound | outbound} {ip-fqdn | fqdn-ip}
6. exit
7. sip ip-fqdn-mapping index ipv4 ip-address fqdn {both-ways | ip-to-fqdn}
8. exit
9. exit
10. exit
11. show sbc sbc-name sbe sip ip-fqdn-mapping

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: Router# config terminal</td>
<td></td>
</tr>
<tr>
<td>Step 2 sbc sbc-name</td>
<td>Enters session border controller (SBC) configuration submode.</td>
</tr>
<tr>
<td>Example: Router(config)# sbc test</td>
<td></td>
</tr>
<tr>
<td>Step 3 sbe</td>
<td>Enters signaling border element (SBE) configuration submode.</td>
</tr>
<tr>
<td>Example: Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td>Step 4 adjacency sip adjacency-name</td>
<td>Enters adjacency SIP configuration submode.</td>
</tr>
<tr>
<td>Example: Router(config-sbc-sbe)# adjacency sip adj1</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td></td>
</tr>
</tbody>
</table>
| `translate (request-uri | to | from) (inbound | outbound) (ip-fqdn | fqdn-ip)` | Configures IP-to-FQDN or FQDN-to-IP translation on SBE.  
| Example:          |         |
| `Router(config-sbc-sbe-adj-sip)# translate request-uri inbound ip-fqdn` |         |
| **Step 6**        |         |
| `exit`            |         |
| Example:          |         |
| `Router(config-sbc-sbe-adj-sip)# exit` |         |
| **Step 7**        |         |
| `sip ip-fqdn-mapping index ipv4 ip-address fqdn (both-ways | ip-to-fqdn)` | Configures SIP IP-to-FQDN mapping on SBE.  
| Example:          |         |
| `Router(config-sbc-sbe)# sip ip-fqdn-mapping 1 ipv4 11.22.33.41 example.sbc1.com both-ways` |         |
| **Step 8**        |         |
| `exit`            |         |
| Example:          |         |
| `Router(config-sbc-sbe)# exit` |         |
| **Step 9**        |         |
| `exit`            |         |
| Example:          |         |
| `Router(config-sbc)# exit` |         |
| **Step 10**       |         |
| `exit`            |         |
| Example:          |         |
| `Router(config)# exit` |         |
| **Step 11**       |         |
| `show sbc sbc-name sbe sip ip-fqdn-mapping` | Displays the currently configured IP-FQDN mappings in the IP-FQDN mapping table.  
| Example:          |         |
| `Router# show sbc test sbe sip ip-fqdn-mapping` |         |

### Configuration Example for SIP IP-FQDN URI Translation

The following example shows how to configure the SIP IP-FQDN URI Translation for Cisco Unified Border Element (SP Edition):

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency sip adj1
Router(config-sbc-sbe-adj-sip)# translate request-uri inbound ip-fqdn
Router(config-sbc-sbe-adj-sip)# exit
Router(config-sbc-sbe)# sip ip-fqdn-mapping 1 ipv4 11.22.33.41 example.sbc1.com both-ways
```
Router(config-sbc-sbe)# exit
Router(config-sbc)# exit
Router(config)# exit
Router# show sbc test sbe sip ip-fqdn-mapping

IP FQDN mappings for SBC service "test"

<table>
<thead>
<tr>
<th>Index</th>
<th>Up?</th>
<th>IP Dir FQDN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Yes</td>
<td>11.22.33.41 &lt;-&gt; example.sbc1.com</td>
</tr>
</tbody>
</table>

* -> = one-way, <-> = both-ways
Router#
SIP Tel URI Support

Cisco Unified Border Element (SP Edition) supports Tel Uniform Resource Identifier (tel URI) in Session Initiation Protocol (SIP) messages, permitting SIP users to set up calls from a SIP IP-phone or SIP User Agent Application to an endpoint in the Public Switched Telephone Network (PSTN). The addition of tel URI to the SIP URI method of connection greatly increases the functionality of Cisco Unified Border Element (SP Edition). SIP can use the tel URI anywhere a URI is allowed, for example, as a Request-URI, along with SIP and SIP URIs.

Note

For Cisco IOS XE Release 2.4 and later, this feature is supported in the unified model.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).

Feature History for SIP Tel URI Support

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 2.4</td>
<td>This feature was introduced on the Cisco IOS XR along with support for the unified model.</td>
</tr>
</tbody>
</table>

Contents

This module contains the following sections:

- Restrictions for SIP Tel URI Support, page 26-1
- Information About SIP Tel URI Support, page 26-2

Restrictions for SIP Tel URI Support

The following is a list of restrictions for SIP tel URI support:

- Cisco Unified Border Element (SP Edition) usually rewrites the domain-name part of the SIP Request-URI header to the configured signaling peer address and port for the outbound adjacency. For example,

  sip:1234567@remote.com

  becomes
Information About SIP Tel URI Support

Local and Global Tel URIs

A tel URI can either be global or local. Global tel URIs are globally unique. Local tel URIs are only valid within a specific local context. For this reason, all local tel URIs must contain the phone-context parameter to specify the context in which they are valid.

The following are examples of global and local tel URIs, respectively.

tel:+358-555-1234567

Note: The separator characters, such as ‘-’ are valid in tel URIs.

tel:1234567;phone-context=+358-555

This URI locates the endpoint with the directory number 1234567 in the context 358-555.

Note: Although the combination of local tel URI and phone-context parameter forms a globally unique identifier, attaching a local tel URI’s phone-context parameter to the tel URI does not necessarily produce a global tel URI. See section 5.1.5 of RFC 3966 for more information.

Tel URI Versus SIP URI

A SIP URI consists of a username and host domain name. A SIP URI uniquely identifies a SIP subscriber but does not necessarily resolve to one particular endpoint on a network. For example,

sip:john@cisco.com

It is also possible to use a directory number as a SIP username and an IP address and port in place of the host domain name. In this case, a SIP URI can uniquely identify an endpoint on a network. For example,

sip:1234567@192.167.1.1:5060

Local tel URIs may or may not contain a domain name in the phone-context parameter. For example,
The Carrier Identification Code Parameter

A Carrier Identification Code (CIC) is a three- or four-digit number used to identify the carrier network in which the destination endpoint of a call is located. It is used by network devices to determine how a call request should be routed between carrier networks. The CIC is often used to specify which carrier network is the current freephone service provider for a freephone number. The current carrier for a given freephone number can be determined by looking up a freephone database.

Tel URIs can include carrier identification codes. For example,

tel: +1-800-234-5678;cic=2345

indicates that the carrier that has been assigned the CIC 2345 is currently the service provider for the freephone number, 1-800-234-5678.

When a network device receives a call request with a tel URI containing a CIC parameter, it will try to route the request according to the value of the CIC parameter. If it cannot route the request, it must decide whether to reject it or continue, ignoring the CIC parameter. If the CIC parameter matches the CIC of the carrier network in which the network device is located, it should route the request based on its local routing policy and strip out the CIC parameter before forwarding the request.

Note
Cisco Unified Border Element (SP Edition) must be explicitly configured to map a CIC value to 0000 in order to strip it out of outbound requests.
SIP Timer

The SIP Timer feature allows the user to configure a number of Session Initiation Protocol (SIP) timers that were hard-coded in the previous releases of Cisco IOS software. The ability to configure SIP timers enables users to improve the interoperability and performance of their devices and network environment.

Note

For Cisco IOS XE Release 2.4 and later, this feature is supported in the unified model only.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).

For a complete description of the commands used in this chapter, refer to the Cisco Unified Border Element (SP Edition) Command Reference: Unified Model at:


For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

Feature History for SIP Timer Functions

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 2.4</td>
<td>This feature was introduced on the Cisco IOS XR along with support for the unified model.</td>
</tr>
</tbody>
</table>

Contents

This module contains the following sections:

- Information About SIP Timer, page 27-1
- How to Configure SIP Timer, page 27-3

Information About SIP Timer

The SIP timer feature allows the user to configure some of the SIP timers that were hardcoded to default values in the previous releases of Cisco IOS software. In the previous releases, Cisco Unified Border Element (SP Edition) used the default SIP timer values recommended by RFC 3261. See Table 27-1.
Information About SIP Timer

Cisco Unified Border Element (SP Edition) allows the user to modify T1, T2 and Timer D, using the `udp-first-retransmit-interval`, `udp-max-retransmit-interval`, and `udp-response-linger-period` commands. You can also use the `invite-timeout` command to choose how long SBC should wait for the remote SIP endpoint to respond to the SBC’s outgoing INVITE or Timer B in an outgoing transaction.

In addition to the SIP protocol-level timers, Cisco Unified Border Element (SP Edition) also allows modification of transport-related timer commands: `tcp-connect-timeout` (how long TCP SYN will wait for the reply) and `tcp-idle-timeout` (how long TCP connection should stay active while idle). Although these timers are transport-level values, Cisco IOS XE Release 2.4 supports these timers in SIP only, but not in H.323, nor H.248.

Note: The incorrect configuration of the SIP timer values may lead to unexpected behavior in certain cases.

<table>
<thead>
<tr>
<th>Timer</th>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>500 ms default</td>
<td>round-trip time (RTT) estimate</td>
</tr>
<tr>
<td>T2</td>
<td>4 s</td>
<td>The maximum retransmit interval for non-INVITE requests and INVITE responses</td>
</tr>
<tr>
<td>T4</td>
<td>5 s</td>
<td>Maximum duration a message will remain in the network</td>
</tr>
<tr>
<td>Timer A</td>
<td>initially T1</td>
<td>INVITE request retransmit interval, for UDP only</td>
</tr>
<tr>
<td>Timer B</td>
<td>64* T1</td>
<td>INVITE transaction timeout timer</td>
</tr>
<tr>
<td>Timer C</td>
<td>&gt; 3 min</td>
<td>Proxy INVITE transaction timeout</td>
</tr>
<tr>
<td>Timer D</td>
<td>&gt; 32 s for UDP</td>
<td>Wait time for response retransmits</td>
</tr>
<tr>
<td></td>
<td>0 s for TCP/Stream Control</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Transmission Protocol (SCTP)</td>
<td></td>
</tr>
<tr>
<td>Timer E</td>
<td>initially T1</td>
<td>non-INVITE request retransmit interval, UDP only</td>
</tr>
<tr>
<td>Timer F</td>
<td>64* T1</td>
<td>non-INVITE transaction timeout timer</td>
</tr>
<tr>
<td>Timer G</td>
<td>initially T1</td>
<td>INVITE response retransmit interval</td>
</tr>
<tr>
<td>Timer H</td>
<td>64* T1</td>
<td>Wait time for ACK receipt</td>
</tr>
<tr>
<td>Timer I</td>
<td>T4 for UDP</td>
<td>Wait time for ACK retransmits</td>
</tr>
<tr>
<td></td>
<td>0 s for TCP/SCTP</td>
<td></td>
</tr>
<tr>
<td>Timer J</td>
<td>64* T1 for UDP</td>
<td>Wait time for non-INVITE request retransmits</td>
</tr>
<tr>
<td></td>
<td>0 s for TCP/SCTP</td>
<td></td>
</tr>
<tr>
<td>Timer K</td>
<td>T4 for UDP</td>
<td>Wait time for response retransmits</td>
</tr>
<tr>
<td></td>
<td>0 s for TCP/SCTP</td>
<td></td>
</tr>
</tbody>
</table>
How to Configure SIP Timer

This section contains the steps for configuring SIP timers.

Configuring SIP Timer

SUMMARY STEPS

1. configure
2. sbc service-name
3. sbe
4. sip timer
5. tcp-connect-timeout interval
6. tcp-idle-timeout interval
7. invite-timeout interval
8. udp-first-retransmit-interval interval
9. udp-max-retransmit-interval interval
10. udp-response-linger-period interval
11. end
12. show sbc service-name sbe sip timers

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc service-name</td>
<td>Enters the mode of an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# sbc mySBC</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Enters the mode of the signaling border element (SBE) function of the SBC.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> sip timer</td>
<td>Enters the mode of the SIP timer function of the SBC.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe)# sip timer</td>
<td></td>
</tr>
</tbody>
</table>
## How to Configure SIP Timer

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 5</strong> tcp-connect-timeout interval</td>
<td>Configures the time (in milliseconds) that SBC waits for a SIP TCP connection to a remote peer to complete before failing that connection. The default timeout interval is 1000 milliseconds.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-sip-tmr)# tcp-connect-timeout 3000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> tcp-idle-timeout interval</td>
<td>Minimum time (in milliseconds) a TCP socket has not processed any traffic, before it is closed. The default is 2 minutes. <strong>Note</strong> The value for this command might not be precise since the idle timers are checked every 12 seconds.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-sip-tmr)# tcp-idle-timeout 30000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> invite-timeout interval</td>
<td>Configures the time (in seconds) that SBC waits for a final response to an outbound SIP INVITE request. The default is 180 seconds. If no response is received during that time, an internal “408 Request Timeout” response is generated and returned to the caller.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-sip-tmr)# invite-timeout 60</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> udp-first-retransmit-interval interval</td>
<td>Configures the time (in milliseconds) that SBC waits for a UDP response or ACK before sending the first retransmission of the relevant signal. If SBC keeps getting no responses, it doubles subsequent retransmission intervals each time until they reach the udp-max-retransmit-interval duration. SBC ceases retransmitting the request and time out the signal if 64 times this duration passes without the receipt of a response/ACK. The default first UDP retransmission interval is 500 milliseconds.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-sip-tmr)# udp-first-retransmit-interval 1000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong> udp-max-retransmit-interval interval</td>
<td>Configures the maximum time interval (in milliseconds) at which SBC will retransmit (see Step 8, udp-first-retransmit-interval above). The default maximum UDP retransmission interval is 4 seconds.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-sip-tmr)# udp-max-retransmit-interval 8000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 10</strong> udp-response-linger-period interval</td>
<td>Configures the time (in milliseconds) for which SBC will retain negative UDP responses to INVITE requests. All subsequent retransmitted responses received within this time will be answered with a negative ACK. Thereafter, any further retransmitted responses are ignored. The default UDP response linger period is 32 seconds.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-sip-tmr)# udp-response-linger-period 10000</td>
<td></td>
</tr>
</tbody>
</table>
### Chapter 27      SIP Timer

#### How to Configure SIP Timer

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 11**  
  `end`  
  **Example:**  
  `Router(config-sbc-sbe-sip-tmr)# end` | Exits the `sip timer` mode and returns to Privileged EXEC mode. |
| **Step 12**  
  `show sbc service-name sbe sip timers`  
  **Example:**  
  `Router# show sbc mysbc sbe sip timers` | Shows the currently configured SIP-related timers. |
Chapter 27  SIP Timer

How to Configure SIP Timer
SIP Configuration Flexibility

Cisco Unified Border Element (SP Edition) offers flexibility in configuring the following features of a Session Initiation Protocol (SIP) adjacency:

- OPTIONS Support
- Rewriting from header on non-REGISTER requests
- Rewriting to header on non-REGISTER requests
- Auto-detecting NAT
- Routing on wildcard domains

For Cisco IOS XR Software Release, this feature is supported in the unified model only.

Feature History for SIP Configuration Flexibility

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XR Software</td>
<td>This feature was introduced on the Cisco IOS XR along with support for</td>
</tr>
<tr>
<td>Release</td>
<td>the unified model.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 3.6S</td>
<td>The Via Header Passthrough feature was added.</td>
</tr>
</tbody>
</table>

Contents

This module contains the following sections:

- Restrictions for Implementing SIP Configuration Flexibility, page 28-2
- Information About SIP Configuration Flexibility, page 28-2
Restrictions for Implementing SIP Configuration Flexibility

The restrictions for implementing SIP configuration flexibility are listed per feature in this chapter.

Information About SIP Configuration Flexibility

This section contains the following subsections:

- OPTIONS Support, page 28-2
- Rewriting From Header on Non-Register Requests, page 28-2

OPTIONS Support

By default, Cisco Unified Border Element (SP Edition) blocks the OPTIONS method from passing through, but users can configure Cisco Unified Border Element (SP Edition) on a per-adjacency basis to pass or block the OPTIONS method by using whitelists and blacklists.

Restrictions for OPTIONS Support

- Cisco Unified Border Element (SP Edition) strips out SDP blocks from messages when it allows the OPTIONS method to pass through. This limits what the SIP endpoints can exchange.
- The SBC-SIG does not send the Accept and Allow headers on any methods, including OPTIONS.
- Cisco Unified Border Element (SP Edition) allows only the 100Rel and Replaces tags of the Supported header to pass through, while the other tags of this header are controlled by whitelists and blacklists.

Rewriting From Header on Non-Register Requests

With this feature, users can configure Cisco Unified Border Element (SP Edition) on a per-adjacency basis to control whether it rewrites the hostport section of the From header on Non-Register Requests to the outbound SIP adjacency address or port. If Cisco Unified Border Element (SP Edition) is configured to allow the From header to pass through without it being rewritten, then Cisco Unified Border Element (SP Edition) allows the entire header to pass through without changing it. The only exception occurs with the Tag parameter; Cisco Unified Border Element (SP Edition) assigns a different value to this parameter before passing it through.

Restrictions for Rewriting From Header on Non-REGISTER Requests

- This feature is not applicable for REGISTER requests.
- This feature may only work in a limited way with the Rewrite-Register feature.
- If the From header contains a Tel URI, then Cisco Unified Border Element (SP Edition) does not rewrite the header since it does not have a hostport.
- Depending on the number of headers, options and SIP whitelist profiles, Cisco Unified Border Element (SP Edition) limits the size of the From header that it allows to pass through to approximately 1000 bytes.

**Rewriting To Header on Non-REGISTER Requests**

The default behavior of Cisco Unified Border Element (SP Edition) is to rewrite the hostport section of the To header on Non-Register Requests to be the outbound SIP adjacency address and port. It also removes any associated parameters. With this feature, users can configure the SBC on a per-adjacency basis to pass the To header through unchanged.

**Auto-detecting NAT**

With the addition of a new configuration field to the SIP adjacency, it is now possible for users to specify if Cisco Unified Border Element (SP Edition) must auto-detect whether a NAT is in use on that adjacency. If Cisco Unified Border Element (SP Edition) is configured to auto-detect NAT, then for each request that it receives, Cisco Unified Border Element (SP Edition) determines whether a NAT is in use for that endpoint. If Cisco Unified Border Element (SP Edition) determines that NAT is in use, then Cisco Unified Border Element (SP Edition) stores the bindings for that request and uses them when sending a response. Additionally, Cisco Unified Border Element (SP Edition) stores and reuses bindings for REGISTER requests for subsequent Dialog-forming and Out-of-dialog requests.

**Restrictions for Auto-detecting NAT**

- Cisco Unified Border Element (SP Edition) can auto-detect NAT only by comparing the Sent-by stopper in the Via header with the remote address and port of the message.
- If the stopper contains a domain name, instead of an IP address, Cisco Unified Border Element (SP Edition) cannot auto-detect whether NAT is in use. In this case, Cisco Unified Border Element (SP Edition) assumes that NAT is in use.
- Auto-detecting NAT is applied only to Out-of-dialog requests or Dialog-forming requests.

**Routing on Wildcard Domains**

Cisco Unified Border Element (SP Edition) routing policy allows you to use the * character in a text domain name match string. This character can match any number of characters in the called address. For example, *domain.com can match both sip1.domain.com and sip2.domain.com.

**Restrictions for Routing on Wildcard Domains**

- You can only specify one wildcard character in a given match string.
- This feature applies only to text domain name match rules, and not to dialed digit match rules.
How to Implement SIP Configuration Flexibility

This section contains the steps for implementing SIP configuration flexibility.

SUMMARY STEPS

1. configure terminal
2. sbc service-name
3. sbe
4. adjacency sip adjacency-name
5. passthrough from header
6. header-name [contact [add [tls-param]] | from{passthrough} | to{passthrough}]
7. nat force-on
8. exit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure terminal</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router# configure terminal</td>
</tr>
<tr>
<td>Enables global configuration mode.</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>sbc service-name</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config)# sbc mysbc</td>
</tr>
<tr>
<td>Enters the mode of an SBC service.</td>
<td></td>
</tr>
<tr>
<td>- Use the service-name argument to define the name of the service.</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>sbe</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc)# sbe</td>
</tr>
<tr>
<td>Enters the mode of the signaling border element (SBE) function of the SBC.</td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>adjacency sip adjacency-name</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe)# adjacency sip sipadj</td>
</tr>
<tr>
<td>Enters the mode of an SBE SIP adjacency.</td>
<td></td>
</tr>
<tr>
<td>- Use the adjacency-name argument to define the name of the SIP adjacency.</td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td>passthrough from header</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-adj-sip)# passthrough from header</td>
</tr>
<tr>
<td>Configures the SIP adjacency to disable From rewrite.</td>
<td></td>
</tr>
</tbody>
</table>
Chapter 28      SIP Configuration Flexibility

Via Header Passthrough

The Via Header Passthrough feature enables the SBC to interoperate with certain devices that use the Via header to authenticate other devices, such as a PBX, that do not support SIP authentication. With the introduction of this feature, the SBC can be configured to function as a SIP proxy that is compliant with RFC 3261 and RFC 3581 in its handling of the received parameter and rport parameter, which are two parameters of the Via header.

This section contains the following topics:

- Restrictions for Via Header Passthrough, page 28-5
- Information About Via Header Passthrough, page 28-6
- How to Configure Via Header Passthrough, page 28-6
- Configuration Example: Via Header Passthrough, page 28-7

Restrictions for Via Header Passthrough

The following are the restrictions for the Via Header Passthrough feature:

- This feature does not support the Topology Hiding feature. After the Via Header Passthrough feature is configured, the data that is passed through the SBC includes information about the topology of the network between the sender and the SBC. If you want to protect the network between the sender and the SBC by using the Topology Hiding feature, do not configure the Via Header Passthrough feature.
- The existing restriction on editing Via headers by using the SIP Message Editing feature is still applicable.

### Step 6

header-name [contact [add [tls-param]] | from (passthrough) | to (passthrough)]

**Example:**

```
Router(config-sbc-sbe-adj-sip)# header-name to passthrough
```

Configures the SIP adjacency to disable To rewrite.

### Step 7

```
nat force-on
```

**Example:**

```
Router(config-sbc-sbe-adj-sip)# nat force-on
```

Configures the SIP adjacency to assume that all endpoints are behind a NAT device. To configure the SIP adjacency to assume that no endpoints are behind a NAT device, use the **nat force-off** command. By default, the SBC autodetects whether the endpoints are behind a NAT device.

### Step 8

```
exit
```

**Example:**

```
Router(config-sbc-sbe-adj-sip)# exit
```

Exits the adj-sip mode and returns to the SBE mode.
Information About Via Header Passthrough

Certain devices use the Via header to authenticate other devices, such as a PBX, that do not support SIP authentication. The Via Header Passthrough feature enables the SBC to interoperate with the devices that use the Via header to authenticate other devices. In releases prior to Release 3.6.0, the SBC would remove the existing Via headers from an incoming SIP message and add its own Via header before forwarding the message. With the introduction of the Via Header Passthrough feature in Release 3.6.0, the SBC can be configured to allow the existing Via headers to pass through and add its own Via header.

When the Via Header Passthrough feature is configured, the SBC adds its own Via header at the top of the stack of Via headers before forwarding the message. If the remote IP address from which the SBC receives the SIP message differs from the IP address specified in the sent-by address of the header, the SBC sets the received parameter to the actual remote IP address before forwarding the message. At the same time, if the SBC receives a SIP message in which the latest entry in the Via header contains the rport parameter with no value set for it, the SBC sets the value of the rport parameter to the source port of the message. In this scenario, the SBC also adds the received parameter to the Via header, regardless of whether the sent-by address in the Via header matches the IP address from which the message was received.

The Via Header Passthrough feature is configured at the SIP adjacency level. To maintain the Via headers on a message routed through the SBC, the Via Header Passthrough feature must be configured on both the inbound adjacency and the outbound adjacency. If this feature is not configured on either one of these adjacencies, the Via headers are removed from the SIP messages that pass through these adjacencies. Note that the SBC adds its own Via header to the outbound SIP message, regardless of whether the Via Header Passthrough feature is configured.

How to Configure Via Header Passthrough

The following procedure shows how to configure the Via Header Passthrough feature.

SUMMARY STEPS

1. configure terminal
2. sbc sbc-name
3. sbe
4. adjacency sip adjacency-name
5. header-name via passthrough inbound
6. header-name via passthrough outbound
7. end
8. show sbc sbc-name sbe adjacencies adjacency-name detail
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters the global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc sbc-name</td>
<td>Enters the configuration mode of an SBC service.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config)# sbc mySbc</td>
<td>• sbc-name—Name of the SBC service.</td>
</tr>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Enters the configuration mode of the signaling border element (SBE) function of the SBC.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> adjacency sip adjacency-name</td>
<td>Enters the mode of an SBE SIP adjacency.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe) # adjacency sip adj1</td>
<td>• adjacency-name—Name of the adjacency.</td>
</tr>
<tr>
<td><strong>Step 5</strong> header-name via passthrough inbound</td>
<td>Specifies that the Via headers on inbound requests for this adjacency must be allowed to pass through.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-adj-sip)# header-name via passthrough inbound</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> header-name via passthrough outbound</td>
<td>Specifies that the Via headers on outbound requests for this adjacency must be allowed to pass through.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-adj-sip)# header-name via passthrough outbound</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> end</td>
<td>Exits the adjacency SIP configuration mode, and returns to the privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-adj-sip)# end</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> show sbc sbc-name sbe adjacencies adjacency-name detail</td>
<td>Shows the configuration details of the specified adjacency.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# show sbc mySBC sbe adjacencies adj1 detail</td>
<td></td>
</tr>
</tbody>
</table>

### Configuration Example: Via Header Passthrough

The following is a sample configuration of the Via Header Passthrough feature:

```
Router(config)# configure terminal
Router(config)# sbc mySbc
Router(config-sbc)# sbe
```
Router(config-sbc-sbe)# adjacency sip adj1
Router(config-sbc-sbe-adj-sip)# header-name via passthrough inbound
Router(config-sbc-sbe-adj-sip)# header-name via passthrough outbound
.
.
.
Router# show sbc mySBC sbe adjacencies adj1 detail

The following is a sample output of the `show sbc mySBC sbe adjacencies adj1 detail` command:

```
Adjacency adj1 (SIP)
  Status: Detached
  Signaling address: 0.0.0.0:default
.
.
  Contact header parameters: Passthrough
  Inbound Via Passthrough: Allowed
  Outbound Via Passthrough: Allowed
.
.
```
SIP Renegotiation

The Cisco Unified Border Element (SP Edition) supports two Session Initiation Protocol (SIP) renegotiation related features:

- **Delta Renegotiation**
  
  The Delta Renegotiation feature determines which SIP renegotiation mode will be used by the session border controller (SBC) when renegotiating media: Delta Renegotiation or Make-Before-Break Renegotiation.

- **Support Renegotiated Call Over NAT**
  
  The Support Renegotiated Call Over NAT feature allows you to ensure that pinholes are preserved for deleted streams so that if the stream is re-enabled, Cisco Unified Border Element (SP Edition) will re-use the same pinhole.

These features significantly reduce the situations in which media ports change mid-call, which provides interoperability and Network Address Translation (NAT) traversal benefits.

---

**Note**

For Cisco IOS XE Release 2.4, the Delta Renegotiation and Support Renegotiated Call Over NAT features are supported in the unified model only.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).

For a complete description of the commands used in this chapter, refer to the Cisco Unified Border Element (SP Edition) Command Reference: Unified Model at:

For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

---

**Feature History for SIP Renegotiation**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 2.4</td>
<td>The Delta Renegotiation and Support Renegotiated Call Over NAT features were introduced on the Cisco IOS XR along with support for the unified model.</td>
</tr>
</tbody>
</table>
Restrictions for Delta Renegotiation

The restrictions for Delta Renegotiation are:

- When Delta Renegotiation mode is in use, stream statistics and Secure Device Provisioning (SDP) billing information will be output at call termination, not at Delta Renegotiation.
- When Delta Renegotiation mode is in use, the following precepts apply:
  - Renegotiation may cause a change in the Differentiated Services Code Point (DSCP) marking policy.
  - The port range depends on the initial incarnation of the stream.
- Under certain scenarios, if the Cisco Unified Border Element (SP Edition) fails over while a Delta Renegotiation is in progress, media resources (such as pinholes and bandwidth allowances) may be unnecessarily allocated.

Information About Delta Renegotiation

The Delta Renegotiation feature determines which SIP renegotiation mode will be used by the Cisco Unified Border Element (SP Edition) when renegotiating media:

- Delta Renegotiation mode
  When the Cisco Unified Border Element (SP Edition) performs a Delta Renegotiation, it retains the existing media pinholes and modifies their variables. Delta Renegotiation mode is used for SIP/H.323 interworked calls and for IP Multimedia Subsystem (IMS) calls.

- Make-Before-Break Renegotiation mode
  When the Cisco Unified Border Element (SP Edition) performs a Make-Before-Break Renegotiation, it creates new pinholes with the proposed media properties, then removes the pre-existing pinholes when the renegotiation completes. These new pinholes temporarily exist in parallel with the existing (old) media pinholes. When the renegotiation completes, Cisco Unified Border Element (SP Edition) deletes the old media pinholes, leaving just the new ones. (Or, if the renegotiation fails, it rolls back to the old state by deleting the new pinholes.)

Delta Renegotiation mode is the default SIP renegotiation mode for all SIP-to-SIP negotiations on the Cisco Unified Border Element (SP Edition) with the following exceptions:

- Change of address family
If the renegotiation changes the address family from IPv4 to IPv6, or vice versa, a new media address is required, and therefore Make-Before-Break Renegotiation mode will be used.

- Mid-call media rerouting

If the renegotiation causes a call to switch between media bypass and non-media bypass mode, the endpoints will perceive a change in the media address, and therefore Make-Before-Break Renegotiation mode will be used.

**Restriction for Support Renegotiated Call Over NAT**

The restriction for the Support Renegotiated Call Over NAT feature is:

- Stream statistics and SDP billing information will be output at call termination, not at Delta Renegotiation.

**Information About Support Renegotiated Call Over NAT**

The Support Renegotiated Call Over NAT feature allows you to ensure that media pinholes are preserved for deleted streams so that if a stream is re-enabled, the Cisco Unified Border Element (SP Edition) will re-use the same pinhole.

This feature is used to avoid de-allocation of a video pinhole in a NAT scenario where Delta Renegotiation mode is in effect and a video transmission is paused. Although the standard SDP protocol when a video transmission is paused is to set the video stream to “a=inactive” (which indicates that SBC should keep the stream allocated), there are known devices that do not set the video stream to “a=inactive” to pause it. Instead, these devices delete the video stream by setting its port to 0. To ensure that the stream remains allocated and the pinhole is preserved even when the SBC receives a port value of 0 during a media stream renegotiation, you can enable the Support Renegotiated Call Over NAT feature.

Use the **media address preserve** command to enable the Support Renegotiated Call Over NAT feature on a per-call basis.

**Configuring Support Renegotiated Call Over NAT**

This section contains the steps to configure the Support Renegotiated Call Over NAT feature, which preserves media pinholes for deleted streams on a per-call basis.

**SUMMARY STEPS**

1. configure
2. sbc service-name
3. sbe
4. cac-policy-set policy-set-id
5. first-cac-table table-name
6. cac-table table-name
7. table-type { policy-set | limit {list of limit tables}}
8. entry entry-id
9. cac-scope \{list of scope options\}
10. [no] media address preserve
11. action cac complete
12. complete
13. active-cac-policy set policy-set-id
14. show sbc service-name sbe cac-policy-set policy-set-id table table-name entry entry-id

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc service-name</td>
<td>Enters the mode of an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# sbc mySbc</td>
<td>Use the <code>service-name</code> argument to define the name of the service.</td>
</tr>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Enters the mode of an SBE entity within an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> cac-policy-set policy-set-id</td>
<td>Enters the mode of CAC policy set configuration within an SBE entity,</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>creating a new policy set if necessary.</td>
</tr>
<tr>
<td>Router(config-sbc-sbe)# cac-policy-set 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> first-cac-table table-name</td>
<td>Configures the name of the first policy table to process when</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>performing the admission control stage of policy.</td>
</tr>
<tr>
<td>Router(config-sbc-sbe-cacpolicy)#</td>
<td></td>
</tr>
<tr>
<td>first-cac-table 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> cac-table table-name</td>
<td>Enters the mode for configuration of an admission control table</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>(creating one if necessary) within the context of an SBE policy set.</td>
</tr>
<tr>
<td>Router(config-sbc-sbe-cacpolicy)#</td>
<td></td>
</tr>
<tr>
<td>cac-table 1</td>
<td></td>
</tr>
</tbody>
</table>
### Step 7

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| `table-type {policy-set | limit (list of limit tables)}` | Configures the table type of a CAC table within the context of an SBC policy set. 

*list of limit tables* can be one of the following values:

- **account**—Compare the name of the account.
- **adj-group**—Compare the name of the adjacency group.
- **adjacency**—Compare the name of the adjacency.
- **all**—No comparison type. All events match this type.
- **call-priority**—Compare with call priority.
- **category**—Compare the number analysis assigned category.
- **dst-account**—Compare the name of the destination account.
- **dst-adj-group**—Compare the name of the destination adjacency group.
- **dst-adjacency**—Compare the name of the destination adjacency.
- **dst-prefix**—Compare the beginning of the dialed digit string.
- **event-type**—Compare with CAC policy event types.
- **src-account**—Compare the name of the source account.
- **src-adj-group**—Compare the name of the source adjacency group.
- **src-adjacency**—Compare the name of the source adjacency.
- **src-prefix**—Compare the beginning of the calling number string.

Features can be enabled or disabled per adjacency group through CAC configuration the same way this is done per individual adjacencies. The adj-group table type matches on either source or destination adjacency group.

When the policy-set keyword is specified, use the **cac-scope** command to configure the scope within each entry at which limits are applied in a CAC Policy Set table.

### Example:

*Router(config-sbc-sbe-cacpolicy-cactable)#

**Example:**

*Router(config-sbc-sbe-cacpolicy-cactable)#

### Step 8

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>entry entry-id</code></td>
<td>Enters the mode to create or modify an entry in an admission control table.</td>
</tr>
</tbody>
</table>

*Example:*

*Router(config-sbc-sbe-cacpolicy-cactable)#

**Example:**

*Router(config-sbc-sbe-cacpolicy-cactable)#

**entry 1**
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 9</strong></td>
<td><strong>cac-scope</strong> <em>(list of scope options)</em></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>Router(config-sbc-sbe-cacpolicy-cactable-entry) # cac-scope src-adjacency</strong></td>
</tr>
<tr>
<td></td>
<td>Choose a scope at which CAC limits are applied within each entry in a Policy Set table. <strong>list of scope options</strong>—Specifies one of the following strings used to match events:</td>
</tr>
<tr>
<td></td>
<td>• account—Events that are from the same account.</td>
</tr>
<tr>
<td></td>
<td>• adjacency—Events that are from the same adjacency.</td>
</tr>
<tr>
<td></td>
<td>• adj-group—Events that are from members of the same adjacency group.</td>
</tr>
<tr>
<td></td>
<td>• call—Scope limits are per single call.</td>
</tr>
<tr>
<td></td>
<td>• category—Events that have same category.</td>
</tr>
<tr>
<td></td>
<td>• dst-account—Events that are sent to the same account.</td>
</tr>
<tr>
<td></td>
<td>• dst-adj-group—Events that are sent to the same adjacency group.</td>
</tr>
<tr>
<td></td>
<td>• dst-adjacency—Events that are sent to the same adjacency.</td>
</tr>
<tr>
<td></td>
<td>• dst-number—Events that have the same destination.</td>
</tr>
<tr>
<td></td>
<td>• global—Scope limits are global</td>
</tr>
<tr>
<td></td>
<td>• src-account—Events that are from the same account.</td>
</tr>
<tr>
<td></td>
<td>• src-adj-group—Events that are from the same adjacency group.</td>
</tr>
<tr>
<td></td>
<td>• src-adjacency—Events that are from the same adjacency.</td>
</tr>
<tr>
<td></td>
<td>• src-number—Events that have the same source number.</td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td><strong>media address preserve</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>Router(config-sbc-sbe-cacpolicy-cactable-entry) # media address preserve</strong></td>
</tr>
<tr>
<td></td>
<td>Ensures that media pinholes are preserved (disabled) for deleted streams.</td>
</tr>
<tr>
<td></td>
<td>• [no]—Allows media pinholes to be deleted for deleted streams.</td>
</tr>
<tr>
<td><strong>Step 11</strong></td>
<td><strong>action cac-complete</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>Router(config-sbc-sbe-cacpolicy-cactable-entry) # action cac complete</strong></td>
</tr>
<tr>
<td></td>
<td>When an event matches, this CAC policy is complete.</td>
</tr>
<tr>
<td><strong>Step 12</strong></td>
<td><strong>complete</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>Router(config-sbc-sbe-cacpolicy) # complete</strong></td>
</tr>
<tr>
<td></td>
<td>Completes the CAC policy set when you have committed the full set.</td>
</tr>
</tbody>
</table>
Chapter 29      SIP Renegotiation

Configuration Example—Support Renegotiated Call Over NAT

The following example enables the Support Renegotiated Call Over NAT feature described in this chapter on a per-call basis

```
sbc mysbc
  sbe
cac-policy-set 1
  first-cac-table 1
cac-table 1
table-type policy-set
ten 1
  media address preserve
  action cac-complete
  complete
active-cac-policy-set 1
```

The following example shows detailed output for the CAC policy set 1, table 1, entry 1, including the “Media Address” field that shows a value of “Preserve,” indicating the Support Renegotiated Call Over NAT feature is enabled

```
Router# show sbc mysbc sbe cac-policy-set 1 table 1 entry 1
SBC Service "mysbc"
Policy set 1 table 1 entry 1
Match value
Action                  Next table
Next-table                      
Max calls                 Unlimited
Max call rate             Unlimited
Max in-call rate          Unlimited
Max out-call rate         Unlimited
Max registrations        Unlimited
Max reg. rate             Unlimited
Max bandwidth             Unlimited
Max channels              Unlimited
Transcoder                Allowed
Caller privacy setting    Never hide
Callee privacy setting    Never hide
Early media               Allowed
Early media direction     Both
Early media timeout       None
Restrict codecs to list   default
Restrict caller codecs to list default
Restrict callee codecs to list default
Media bypass              Allowed
```

### Command or Action | Purpose
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 13 active cac-policy-set policy-set-id</td>
<td>Sets the active CAC policy set within an SBE entity.</td>
</tr>
<tr>
<td>Example: Router(config-sbc-sbe)# active cac-policy-set 1</td>
<td></td>
</tr>
<tr>
<td>Step 14 show sbc service-name sbe cac-policy-set policy-set-id table table-name entry entry-id</td>
<td>Lists detailed information for a given entry in a CAC policy table, including whether the Support Renegotiated Call Over NAT feature is enabled. When this feature is enabled, the “Media Address” field shows a value of “Preserve.”</td>
</tr>
<tr>
<td>Example: Router# show sbc mysbc sbe cac-policy-set 1 table 1 entry 1</td>
<td></td>
</tr>
</tbody>
</table>
### Configuration Example—Support Renegotiated Call Over NAT

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRTP Transport</td>
<td>Trusted-Only (by default)</td>
</tr>
<tr>
<td>Callee hold setting</td>
<td>Standard</td>
</tr>
<tr>
<td>Caller hold setting</td>
<td>Standard</td>
</tr>
<tr>
<td>Media Address</td>
<td>Preserve</td>
</tr>
<tr>
<td>Renegotiate</td>
<td>Delta</td>
</tr>
<tr>
<td>Number of calls rejected by this entry</td>
<td>0</td>
</tr>
</tbody>
</table>
100rel Interworking Support

Cisco Unified Border Element (SP Edition) provides support for 100rel (SIP Provisional Message Reliability) interworking. This feature provides support to resolve the interoperability problem of inconsistent support for SIP reliable provisional responses encountered when SBC works with different SIP networks.

SIP defines two types of responses: provisional and final. Final responses (2xx-6xx) convey the result of the request processing and are sent reliably. SIP provisional responses (1xx) do not have an acknowledgement system so they are not reliable. There are certain scenarios in which the provisional SIP responses (1xx) must be delivered reliably. For example in a SIP/PSTN interworking scenario it is crucial that the 180 and 183 messages are not dropped. The use of the Provisional Response ACKnowledgment (PRACK) method enables reliability to be offered to SIP provisional responses.

The 100rel option is used to indicate that the reliable provisional responses are supported or required, and the PRACK message is used to acknowledge receipt of a reliable provisional response.

---

**Note**

This feature is supported in the unified model for Cisco IOS XE Release 2.5 and later.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).

**Feature History for 100rel Interworking Support**

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 2.5</td>
<td>This feature was introduced on the Cisco IOS XR.</td>
</tr>
</tbody>
</table>

**Contents**

This module contains the following sections:

- Restrictions for 100rel Interworking Support, page 30-2
- Information About 100rel Interworking Support, page 30-2
- Configuring 100rel Interworking Support, page 30-5
Restrictions for 100rel Interworking Support

The following restrictions apply when you configure the 100rel interworking support on the Cisco Unified Border Element (SP Edition):

- If late to early media interworking is required, the callee must support reliable provisional responses, and the scenario shown in Figure 30-1 must not be configured.
- The 100rel interworking allows only one offer exchange on PRACK messages for each INVITE transaction.
- The 100rel interworking is configured on the adjacency facing the network that requires 100rel support:
  - The Cisco Unified Border Element (SP Edition) configuration must be set up on the inbound adjacency of the applicable call to act as a PRACK User Agent Server (UAS) during 100rel interworking.
  - The Cisco Unified Border Element (SP Edition) configuration must be set up on the outbound adjacency of the applicable call to act as a PRACK User Agent Client (UAC) during 100rel interworking.
- The SIP uses provisional responses to avoid transaction time-out while the final response is outstanding, and reduces the frequency of these progress responses when they are sent reliably. This allows a B2BUA that receives unreliable progress responses and sends reliable progress responses to send progress responses less frequently than it receives them. Cisco Unified Border Element (SP Edition) does not attempt to do this, it simply forwards provisional responses when they are received (subject to any configured header filtering rules).

Information About 100rel Interworking Support

The 100rel interoperability feature performs the following functions on individual SIP adjacencies:

- Strips the 100rel option from incoming SIP requests.
- Sends reliable provisional responses to the caller UAC even when the responses from the called UAS are unreliable.
- Receives reliable provisional responses from the called UAS even if the caller UAC does not support them.
- Adds support for the 100rel option to outgoing SIP requests.

Figure 30-1 shows SBC acting as UAS, and Figure 30-2 shows SBC acting as UAC.
Figure 30-1 SBC Acting as UAS

Caller UAC

INVITE Supported: 100rel

100 Trying

180 Ringing Require: 100rel

PRACK

200 OK (PRACK)

200 OK (INVITE)

ACK

Called UAS

INVITE

100 Trying

180 Ringing

200 OK (INVITE)

ACK

SBC strips “Supported: 100rel” from the INVITE request

SBC is configured to send reliable provisional responses

Final response (reliable)
Information About 100rel Interworking Support

Caller UAC Requires 100rel

Cisco Unified Border Element (SP Edition) sets up calls from a network that requires 100rel, but needs to be routed to networks that do not support 100rel. To facilitate this, SBC strips the 100rel option from Supported and Require headers in SIP requests. After stripping the 100rel option, SBC still sends reliable provisional responses with a “Require: 100rel” header if required. Cisco Unified Border Element can also be configured to send reliable provisional responses to requests that include a "Supported: 100rel" header when such requests do not include a "Require: 100rel" header and responses are received as unreliable provisional responses.

Send Reliable Responses if Required

If a SIP request includes a “Require: 100rel” header and SBC strips the 100rel option then it must send provisional responses as reliable provisional responses with a “Require: 100rel” header. In this case the called UAS sends unreliable provisional responses because SBC has stripped the 100rel option from the request.

Send Reliable Responses if Supported

If a SIP request includes a “Supported: 100rel” header then SBC must send reliable provisional responses to the caller UAC even when the SIP request does not include a “Require: 100rel” header and the called UAS sends unreliable provisional responses.
Calle UAS Requires 100rel

Cisco Unified Border Element (SP Edition) sends requests to networks that require 100rel from networks that do not. To facilitate this, the following functions are required:

Advertise Support for 100rel
SIP Requests passing through SBC should have “Supported: 100rel” header added to them.

Remove 100rel from Responses
If SBC advertises support for 100rel, then it also ensures that the non-PRACK network receives non-100rel messages.

Configuring 100rel Interworking Support

Cisco Unified Border Element (SP Edition) requires following configurations to enable 100rel interworking support:

- The configuration is applied to SIP adjacencies.
- At the incoming side, two flags are configured to indicate,
  - whether to strip 100rel option from Supported and Require headers in the incoming SIP INVITE request.
  - whether to enable 100rel interworking if incoming SIP INVITE request contains “Supported:100rel” header.
- At the outgoing side, two flags are configured to:
  - add “Supported:100rel” in the outgoing SIP INVITE request.
  - add “Require:100rel” in the outgoing SIP INVITE request.

This section contains the steps to configure the 100rel interworking support.

SUMMARY STEPS

1. configure terminal
2. sbc sbc-name
3. sbe
4. adjacency sip adjacency-name
5. 100rel inbound {strip | support}
6. 100rel outbound {require-add | support-add}
7. end
8. show sbc sbc-name sbe adjacencies {adjacency-name} [detail]
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# config terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc sbc-name</td>
<td>Enters session border controller (SBC) configuration submode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config)# sbc test</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Enters signaling border element (SBE) configuration submode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> adjacency sip adjacency-name</td>
<td>Enters adjacency SIP configuration submode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe)# adjacency sip adj1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> 100rel inbound (strip</td>
<td>support)</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-adj-sip)# 100rel inbound strip</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-adj-sip)# 100rel inbound support</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> 100rel outbound (require-add</td>
<td>support-add)</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-adj-sip)# 100rel outbound require-add</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-adj-sip)# 100rel outbound support-add</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> end</td>
<td>Returns to the privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe)# end</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc)#</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> show sbc sbc-name sbe adjacencies (adjacency-name) [detail]</td>
<td>Lists the adjacencies configured on signaling border elements (SBEs).</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# show sbc test sbe adjacencies adj1 detail</td>
<td></td>
</tr>
</tbody>
</table>
Customized System Error Messages

Cisco Unified Border Element (SP Edition) supports customized system error messages.

Feature History for Implementing SNMP

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 3.1S</td>
<td>Customized System Error Messages feature was introduced.</td>
</tr>
</tbody>
</table>

Contents

- Information About Customized System Error Messages, page 31-1
- Configuring Customized System Error Messages, page 31-4
- Configuration Example of Implementing Customized System Error Message, page 31-6

Information About Customized System Error Messages

SBC provides the ability to map internal system error-codes to SIP status-codes, and gives system administrators the ability to add a customer configured Reason: header into the response.

System administrators can map and customize error messages in user-defined error profiles. The following types of existing SIP error codes can be mapped and customized in user-defined error profiles:

- Call Admission Control (CAC)
- Number Analysis (NA)
- Routing Errors (RTG)

A default error profile is automatically created and attached to SIP adjacencies during SBE configuration. The default error profile can be modified, but cannot be deleted.

User-defined error-profiles are added to the existing SIP profiles and can be attached to adjacencies. Errors are identified by a cause and an optional a sub-cause. If no sub-cause is entered, all possible sub-causes are mapped to that cause.

Each cause/sub-cause combination can be mapped by the user to any SIP status-code in the range between 400 and 699.
When an internal error is generated, the system checks for a configured cause/sub-cause mapped to that error. The system first checks the adjacency for specific error-profile, then it checks the default profile for an equivalent error mapping. If no match is found, the existing internal error message is returned. If a configured error profile is found, it overwrites the internal error message.

A user-defined error-profile contains the following elements:

- **Cause**
- **Sub-cause**
- **Status-code**
- **Reason**

### Cause

In an error profile, the cause of an internal error is specified, using the `cause` command to select one of the following available CLI causes:

- **cac-in-call-msg-rate**—cac: The rate of mid-call messages has exceeded a maximum configured limit
- **cac-max-bandwidth**—cac: The bandwidth used has exceeded a maximum configured limit
- **cac-max-call-rate**—cac: Call setup rate exceeded a maximum configured limit
- **cac-max-channels**—cac: The number of media channels used has exceeded a maximum limit
- **cac-max-num-calls**—cac: The number of calls has exceeded a maximum limit
- **cac-max-reg**—cac: The number of registrations has exceeded a maximum configured limit
- **cac-max-reg-rate**—cac: The rate of registrations has exceeded a maximum configured limit
- **cac-max-updates**—cac: The number of call updates has exceeded the configured limit
- **cac-out-call-msg-rate**—cac: The rate of out of dialogue messages has exceeded a maximum configured limit
- **cac-rtp-disallowed**—cac: Disallowing rtp caused the call to fail
- **cac-srtp-disallowed**—cac: Disallowing srtp caused the call to fail
- **cac-srtp-rtp-interwork**—cac: call failed due to srtp to rtp interworking disallowed
- **enum-failure**—ENUM processing encountered an error
- **max-media-streams**—An offer cannot be reduced to meet the maximum number of media streams
- **mg-srtp-unsupported**—No MG was found which can support srtp
- **na-invalid-address**—na: Number validation failure
- **no-acceptable-codec**—No acceptable codec can be found for an offer
- **rtg-max-routes-tried**—rtg: The maximum number of routing attempts exceeded
- **rtg-no-route-found**—rtg: Routing failed to find a route
- **rtg-route-unavailable**—rtg: The route selected by call-policy is unavailable
- **srtp-general-error**—srtp general error
- **sub-media-bearer-chan-fail**—subscriber media bearer channel has failed mid-call
• **sub-media-bearer-chan-rej**—subscriber media bearer channel has rejected during setup or renegotiation
• **sub-sig-bearer-chan-fail**—subscriber signaling bearer channel is unavailable

**Sub-cause**

After the cause is selected, the sub-cause can then be selected optionally.

To see a list of the available sub-causes for each cause, use the question mark (?) online help function after you have selected the cause. The following list shows all available sub-causes:

• **na-dst-number**—Destination number based analysis
• **na-src-adjacency**—Source adjacency based analysis
• **na-src-account**—Source account based analysis
• **na-sub-category**—Subscriber category based analysis
• **na-carrier-id**—Carrier identification code based analysis
• **na-src-number**—Source number based analysis
• **na-no-src-number**—No source number present for source number based analysis
• **rtg-src-address**—Source address based routing
• **rtg-dst-address**—Destination address based routing
• **rtg-src-adjacency**—Source adjacency based routing
• **rtg-src-account**—Source account based routing
• **rtg-category**—Category based routing
• **rtg-sub-category**—Subscriber category based routing
• **rtg-src-domain**—Source domain based routing
• **rtg-dst-domain**—Destination domain based routing
• **rtg-time**—Time based routing
• **rtg-dst-tgid**—Destination trunk group Identifier based routing
• **rtg-src-tgid**—Source trunk group identifier based routing
• **rtg-carrier-id**—Carrier identification code based routing
• **rtg-round-robin**—Round robin based routing
• **rtg-least-cost**—Least cost based routing
• **cac-unknown**—Unknown call admission control error
• **cac-per-call-scope**—Call admission control call scope error
• **cac-src-number-scope**—Call admission control source number scope error
• **cac-downstream-scope**—Call admission control downstream scope attribute error
• **cac-upstream-scope**—Call admission control upstream scope attribute error
• **sub-rx-reg-bearer-loss**—Failed to route to a subscriber because the Rx session for the subscriber registration suffered loss of bearer
• **sub-rx-reg-bearer-rel**—Failed to route to a subscriber because the rx session for the subscriber registration suffered release of bearer
• **sub-rx-reg-bearer-term**—Failed to route to a subscriber because the rx session for the subscriber registration was terminated
Configuring Customized System Error Messages

Status-code
The SIP status-code numbers range from 400 to 699. A SIP status code can be mapped to a selected cause/sub-cause, using the cause command.

Reason
The reason field allows system administrators to optionally configure a SIP "Reason:" header, which is inserted into the error response and displayed when an error occurs. The configured reason header must conform to the syntax rules defined in RFC 3326.

Configuring Customized System Error Messages

Use the following procedure to configure custom error messages.

SUMMARY STEPS
1. configure terminal
2. sbc sbc-name
3. sbe
4. sip error-profile error-profile-name
5. description description
6. cause cause [sub-cause sub-cause] status-code status-code [reason reason]
7. exit
8. adjacency sip adjacency-name
9. error-profile outbound profile-name
10. end
11. show sbc sbc-name sbe sip error-profile

- sub-rx-media-policy-rej—Rx session for a call was rejected for policy reasons (for example, unsupported media)
- sub-rx-media-error—Rx session for a call was rejected for non-policy reasons (for example, service unavailable)
- sub-rx-reg-bearer-loss—Rx session for a call suffered loss of bearer
- sub-rx-reg-bearer-rel—Rx session for a call suffered release of bearer
- sub-rx-reg-bearer-term—Rx session for a call was terminated
- enum-resource—enum - encountered a resource shortage
- enum-dst-not-number—enum - destination address which was not a telephone number
- enum-unknown-number—enum - unable to resolve a telephone number
- enum-interface-failure—enum - failed in the enum interface
- enum-regex-error—enum - failed because a regex in a NAPTR record was invalid
# Chapter 31 Customized System Error Messages

## Configuring Customized System Error Messages

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Router# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td><code>sbc sbc-name</code></td>
<td>Creates the SBC service on Cisco Unified Border Element (SP Edition) and enters into SBC configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Router(config)# sbc SBC1</code></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td><code>sbe</code></td>
<td>Enters the mode of the signaling border element (SBE) function of the SBC.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Router(config-sbc)# sbe</code></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td><code>sip error-profile error-profile-name</code></td>
<td>Creates an error profile and enters error profile configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Router(config-sbc-sbe)# sip error-profile IN_Err_Profile_1</code></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td><code>description description</code></td>
<td>Adds a description to an error profile.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Router(config-sbc-sbe-sip-err)# description call rate error</code></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td><code>cause cause [sub-cause sub-cause] status-code status-code [reason reason]</code></td>
<td>Configures the cause, sub-cause, status-code, and reason of an internal error for an error profile.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Router(config-sbc-sbe-sip-err)# cause cac-max-reg status-code 553 reason &quot;SIP ;cause=503 ;text=\&quot; Exceed the max reg num\&quot;&quot;</code></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td><code>exit</code></td>
<td>Exits to the previous mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Router(config-sbc-sbe-err)# exit</code></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td><code>adjacency sip adjacency-name</code></td>
<td>Enters the mode of an SBE SIP adjacency.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Router(config-sbc-sbe)# adjacency sip Adj_1</code></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td><code>error-profile outbound profile-name</code></td>
<td>Configures an existing error profile as the outbound SIP error profile.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Router(config-sbc-sbe-adj-sip)# error-profile outbound OUT_Err_Profile_1</code></td>
<td></td>
</tr>
</tbody>
</table>
### Configuration Example of Implementing Customized System Error Message

The following example shows how to configure custom error messages:

```plaintext
Router# configure terminal
Router(config)# sbc SBC1
Router(config-sbc)# sbe
Router(config-sbc-sbe)# sip error-profile Error_Profile_1
Router(config-sbc-sbe-sip-err)# description call rate error
Router(config-sbc-sbe-sip-err)# cause cac-max-reg status-code 553 reason "SIP ;cause=503; text=" Exceed the max reg num"
Router(config-sbc-sbe-adj-sip)# adjacency sip Adj_1
Router(config-sbc-sbe-adj-sip)# error-profile outbound OUT_Err_Profile_1
Router(config-sbc-sbe-enum-entry)# end
Router# show sbc SBC1 sbe sip error-profile
```

### Command or Action

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 10</strong> end</td>
<td>Exits configuration mode and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Step 11</strong> show sbc sbc-name sbe sip error-profile</td>
<td>Displays the configuration information for all error profiles.</td>
</tr>
</tbody>
</table>

**Example:**

```
Router(config-sbc-sbe-enum-entry)# end
```

Exits configuration mode and returns to privileged EXEC mode.

**Example:**

```
Router# show sbc SBC1 sbe sip error-profile
```

Displays the configuration information for all error profiles.
BFCP Support

Binary Floor Control Protocol (BFCP), defined in RFC 4582, is a protocol for controlling the access to the media resources in a conference.

Cisco Unified Border Element (SP Edition) was earlier known as Integrated Session Border Controller. It is referred to in this document as the session border controller (SBC).

For a complete description of the commands used in this chapter, refer to the *Cisco Unified Border Element (SP Edition) Command Reference: Unified Model* at:


For information about all the Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

Feature History for BFCP Support

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE 3.3S Release</td>
<td>This feature was introduced on the Cisco ASR 1000 Series Routers.</td>
</tr>
</tbody>
</table>

Contents

This chapter contains the following sections:

- Prerequisite for BFCP Support, page 32-1
- Restrictions for BFCP Support, page 32-2
- Information About BFCP Support, page 32-2
- Configuring BFCP Support, page 32-3
- Configuration Example of BFCP Support, page 32-9

Prerequisite for BFCP Support

Following is the prerequisite pertaining to the BFCP Support feature:

- The SBC must pass through the $b=CT$ line and the $a=rtcp-fb:* nack pli$ RTCP feedback information included in the Session Description Protocol (SDP).
Restrictions for BFCP Support

Following are the restrictions pertaining to the BFCP Support feature:

- The SBC treats a generic media stream the same way it treats other media streams. Therefore, a call is released only if all the media streams are reported as being inactive. The Media Packet Forwarder (MPF) media timer is processed in the same way as the other voice or video streams pertaining to the BFCP stream.
- A BFCP media stream and a generic media stream do not have a bandwidth specified. Therefore, it can be policed only by the MPF, and not the Call Admission Control (CAC) total bandwidth limits.
- The SBC does not support the generic TCP streams or BFCP over TCP. Therefore, a request to add a TCP stream to the generic media stream configuration gets rejected.
- H.323 calls or H.323-SIP interworking calls are not supported.

Information About BFCP Support

The BFCP Support feature supports BFCP over UDP in the SBC by configuring BFCP as a recognized generic media stream that can be forwarded using the best-effort traffic class.

Generic media streams are media streams in which the media (m)-line definition uses * instead of a codec list, for example, m=application port UDP/BFCP *. By default, the SBC cuts these m-lines out of the SDP offers and replies by setting the port to zero. These media lines carry no bandwidth information and therefore, cannot be policed against CAC limits, denial of service, or media theft attacks of the SBC.

The BFCP Support feature introduces the best-effort traffic class that allows policing of these media lines in the media forwarder.

The SBC can be configured to accept specific generic media streams. After this, the accepted generic media streams are added to the best-effort traffic class. MPF implementation supports the best-effort traffic class by policing the actual usage of the aggregate of these streams.

Best-Effort Traffic Class

Prior to Cisco IOS XE Release 3.3S, the media streams had their bandwidth specified for audio and video streams, or were not subjected to any policing, such as T120. From Cisco IOS XE Release 3.3S, the SBC is configured to accept arbitrary type and number of generic media streams. Some of the BFCP streams can now have low bandwidth protocol messages. The best-effort traffic class simplifies packet policing because it allows a media forwarder to handle such streams cumulatively. The best-effort traffic class rate limit is 1Mbps cumulatively.
Configuring BFCP Support

This section describes how to configure the BFCP Support feature on the SBC.

**SUMMARY STEPS**

1. configure terminal
2. sbc service-name
3. `sbe`
4. `stream-list stream-list-name`
5. `description description`
6. `generic-stream media-type {application | message} transport udp protocol protocol-name`
7. `exit`
8. `cac-policy-set policy-set-id`
9. `cac-table table-name`
10. `table-type {policy-set | limit {list of limit tables}}`
11. `entry entry-id`
12. `action [next-table goto-table-name | cac-complete]`
13. `generic-stream caller generic-stream-list`
14. `generic-stream callee generic-stream-list`
15. `match-value key`
16. `exit`
17. `exit`
18. `complete`
19. `end`
20. `show sbc service-name sbe stream-list`
21. `show sbc service-name sbe cac-policy-set id table name entry entry`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure terminal</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router# configure terminal</td>
</tr>
<tr>
<td>Enables the global configuration mode.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>sbc sbc-name</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config)# sbc mysbc</td>
</tr>
<tr>
<td>Enters the SBC service mode. Use the <code>sbc-name</code> argument to define the name of the service.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>sbe</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc)# sbe</td>
</tr>
<tr>
<td>Enters the SBE entity mode within an SBC service.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>stream-list stream-list-name</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe)# stream-list my_stream</td>
</tr>
<tr>
<td>Configures a stream list and enters the stream list configuration mode.</td>
<td></td>
</tr>
<tr>
<td>• <code>stream-list-name</code>—The name of the stream list. The stream list name can be up to 30 characters.</td>
<td></td>
</tr>
</tbody>
</table>
**Chapter 32  BFCP Support**

**Configuring BFCP Support**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td><code>description</code> description</td>
<td>Configures descriptive text for the stream list.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> Router(config-sbc-sbe-stream-list)# description &quot;This is my first stream list&quot;</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>`generic-stream media-type application</td>
<td>message) transport udp protocol protocol-name`</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> Router(config-sbc-sbe-stream-list)# generic-stream media-type application transport udp protocol BFCP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- <code>application</code>—Specifies <code>application</code> as media type for the generic stream.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- <code>message</code>—Specifies <code>message</code> as media type for the generic stream.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- <code>transport</code>—Configures the transport protocol for the generic stream.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- <code>udp</code>—Specifies the UDP protocol for the generic stream.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- <code>protocol</code>—Specifies the protocol name for the generic stream.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- <code>protocol-name</code>—The protocol name for the generic stream. The protocol name is case sensitive.</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td><code>exit</code></td>
<td>Exits from the stream list configuration mode and enters the SBE configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> Router(config-sbc-sbe-stream-list)# exit</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td><code>cac-policy-set policy-set-id</code></td>
<td>Enters the CAC policy set configuration mode within an SBE entity, creating a new policy set, if necessary.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> Router(config-sbc-sbe)# cac-policy-set 2</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td><code>cac-table table-name</code></td>
<td>Enters the admission control table configuration mode (creating one, if necessary) within the context of an SBE policy set.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> Router(config-sbc-sbe-cacpolicy)# cac-table 2</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>`table-type policy-set</td>
<td>limit (list of limit tables)`</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> Router(config-sbc-sbe-cacpolicy-cactable)# table-type src-adjacency</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td><code>entry entry-id</code></td>
<td>Enters the CAC table entry mode to create or modify an entry in an admission control table.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> Router(config-sbc-sbe-cacpolicy-cactable)# entry 1</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 12</td>
<td>`action [next-table goto-table-name</td>
<td>Configures the action to be performed after the entry, in an admission control table. Possible actions are:</td>
</tr>
<tr>
<td></td>
<td>cac-complete]`</td>
<td>• Identifies the next CAC table to be processed using the <code>next-table</code> keyword and the <code>goto-table-name</code> argument.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Stops the processing for the scope using the <code>cac-complete</code> keyword.</td>
</tr>
</tbody>
</table>
|        | Example:                           | **Router(config-sbc-sbe-cacpolicy-cactable-entry)`
|        | # action cac-complete              | Example: **`# action cac-complete`**                                                                                                   |
| Step 13 | `generic-stream caller generic-stream-list` | Configures the generic media stream list settings for a caller.                                                                        |
|        | Example:                           | **Router(config-sbc-sbe-cacpolicy-cactable-entry)`
|        | # generic-stream caller my-stream  | Example: **`# generic-stream caller my-stream`**                                                                                      |
| Step 14 | `generic-stream callee generic-stream-list` | Configures the generic media stream list settings for a callee.                                                                      |
|        | Example:                           | **Router(config-sbc-sbe-cacpolicy-cactable-entry)`
|        | # generic-stream callee my-stream  | Example: **`# generic-stream callee my-stream`**                                                                                       |
| Step 15 | `match-value key`                  | Configures the match value of an entry in a CAC limit table.                                                                          |
|        | Example:                           | **Router(config-sbc-sbe-cacpolicy-cactable-entry)`
|        | # match-value SIP-adj-test         | Example: **`# match-value SIP-adj-test`**                                                                                              |
| Step 16 | `exit`                             | Exits from the CAC table entry configuration mode and enters the CAC table mode.                                                        |
|        | Example:                           | **Router(config-sbc-sbe-cacpolicy-cactable-entry)`
|        | # exit                             | Example: **`# exit`**                                                                                                                   |
| Step 17 | `exit`                             | Exits from the CAC table configuration mode and enters the CAC policy set configuration mode.                                            |
|        | Example:                           | **Router(config-sbc-sbe-cacpolicy-cactable)`
|        | # exit                             | Example: **`# exit`**                                                                                                                   |
| Step 18 | `complete`                         | Completes the CAC policy set after you have committed the complete set.                                                               |
|        | Example:                           | **Router(config-sbc-sbe-cacpolicy)`
|        | `complete`                         | Example: **`# complete`**                                                                                                               |
| Step 19 | `end`                              | Exits from the CAC policy set configuration mode and enters the Privileged EXEC mode.                                                   |
|        | Example:                           | **Router(config)`
|        | `end`                              | Example: **`# end`**                                                                                                                   |
### Command or Action

<table>
<thead>
<tr>
<th>Step 20</th>
<th>show sbc sbc-name sbe stream-list</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Router# show sbc mysbc sbe stream-list my-stream</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 21</th>
<th>show sbc service-name sbe cac-policy-set id table name entry entry</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Router# show sbc mysbc sbe cac-policy-set 1 table MyTable entry 1</td>
</tr>
</tbody>
</table>

---

The following example shows the output of the `show sbc sbe stream-list` command:

```
Router# show sbc mysbc sbe stream-list
SBC Service "mysbc"

Stream list: my-stream
   Description This is my first stream list
   Media-type application Transport udp protocol Streambased
   Media-type message Transport udp protocol BFCP
```

The following example shows the output of the `show sbc sbe cac-policy-set table entry` command:

```
Router# show sbc mysbc sbe cac-policy-set 25 table 2 entry 1
SBC Service "Mysbc"
CAC Averaging period 1: 60 sec
CAC Averaging period 2: 0 sec

CAC Policy Set 25
Global policy set: No
Description:
   First CAC table:
   First CAC scope: global

Table name: 2
Description:
   Table type: limit src-adjacency
   Total call setup failures (due to non-media limits): 0

Entry 1
   Match value:
   Match prefix length: 0
   Action: CAC complete
   Number of call setup failures (due to non-media limits): 0
   No. of registrations rejected (due to registration limits): 0

   Max calls per scope: Unlimited
   No. of events rejected due to Max Call Limit: 0
   Max reg. per scope: Unlimited
   No. of events rejected due to Max Reg limit: 0
   Max channels per scope: Unlimited
   Max updates per scope: Unlimited
   Max bandwidth per scope: Unlimited
```
Chapter 32  BFCP Support

Configuring BFCP Support

2

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Averaging-period 1</th>
<th>Averaging-period 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max call rate per scope:</td>
<td>Unlimited</td>
<td>Unlimited</td>
</tr>
<tr>
<td>No. of events rejected due to Max call rate:</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Max reg. rate per scope:</td>
<td>Unlimited</td>
<td>Unlimited</td>
</tr>
<tr>
<td>No. of events rejected due to Max reg rate:</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Max in-call message rate:</td>
<td>Unlimited</td>
<td>Unlimited</td>
</tr>
<tr>
<td>No. of events rejected due to Max in-call rate</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Max out-call message rate:</td>
<td>Unlimited</td>
<td>Unlimited</td>
</tr>
<tr>
<td>No. of events rejected due to Max Out call rate</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Timestamp when the rejection counts were last reset:</td>
<td>2011/01/03 22:29:40</td>
<td></td>
</tr>
<tr>
<td>Early media:</td>
<td>Allowed</td>
<td>Both</td>
</tr>
<tr>
<td>Early media timeout:</td>
<td>None</td>
<td>Transcoder per scope: Allowed</td>
</tr>
<tr>
<td>Callee Bandwidth-Field:</td>
<td>None</td>
<td>Caller Bandwidth-Field: None</td>
</tr>
<tr>
<td>Media bypass:</td>
<td>Allowed</td>
<td>Asymmetric Payload Type: Not Set</td>
</tr>
<tr>
<td>Renegotiate Strategy:</td>
<td>Delta</td>
<td></td>
</tr>
<tr>
<td>SRTP Transport:</td>
<td>Trusted-Only (by default)</td>
<td></td>
</tr>
<tr>
<td>Caller hold setting:</td>
<td>Standard</td>
<td></td>
</tr>
<tr>
<td>Callee hold setting:</td>
<td>Standard</td>
<td></td>
</tr>
<tr>
<td>Caller limited-privacy-service:</td>
<td>Never hide identity</td>
<td></td>
</tr>
<tr>
<td>Callee limited-privacy-service:</td>
<td>Never hide identity</td>
<td></td>
</tr>
<tr>
<td>Caller privacy-service:</td>
<td>Not set</td>
<td></td>
</tr>
<tr>
<td>Callee privacy-service:</td>
<td>Not set</td>
<td></td>
</tr>
<tr>
<td>Caller edit-privacy-request:</td>
<td>Not set</td>
<td></td>
</tr>
<tr>
<td>Callee edit-privacy-request:</td>
<td>Not set</td>
<td></td>
</tr>
<tr>
<td>Caller edit-privacy-request sip strip:</td>
<td>Not set</td>
<td></td>
</tr>
<tr>
<td>Callee edit-privacy-request sip strip:</td>
<td>Not set</td>
<td></td>
</tr>
<tr>
<td>Caller edit-privacy-request sip insert:</td>
<td>Not set</td>
<td></td>
</tr>
<tr>
<td>Callee edit-privacy-request sip insert:</td>
<td>Not set</td>
<td></td>
</tr>
<tr>
<td>Caller voice QoS profile:</td>
<td>Default</td>
<td></td>
</tr>
<tr>
<td>Callee voice QoS profile:</td>
<td>Default</td>
<td></td>
</tr>
<tr>
<td>Caller video QoS profile:</td>
<td>Default</td>
<td></td>
</tr>
<tr>
<td>Callee video QoS profile:</td>
<td>Default</td>
<td></td>
</tr>
<tr>
<td>Caller sig QoS profile:</td>
<td>Default</td>
<td></td>
</tr>
<tr>
<td>Callee sig QoS profile:</td>
<td>Default</td>
<td></td>
</tr>
<tr>
<td>Caller inbound SDP policy:</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>Callee inbound SDP policy:</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>Caller outbound SDP policy:</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>Callee outbound SDP policy:</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>SDP Media Profile :</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>Caller Generic Stream :</td>
<td>my-stream</td>
<td></td>
</tr>
<tr>
<td>Callee Generic Stream :</td>
<td>my-stream</td>
<td></td>
</tr>
<tr>
<td>Caller media disabled:</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>Callee media disabled:</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>Caller unsignaled secure media:</td>
<td>Not Allowed</td>
<td></td>
</tr>
<tr>
<td>Callee unsignaled secure media:</td>
<td>Not Allowed</td>
<td></td>
</tr>
<tr>
<td>Caller response downgrade support:</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Callee response downgrade support:</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Caller retry rtp support:</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Callee retry rtp support:</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Resend sdp answer in 200ok:</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Caller tel-event payload type:</td>
<td>Default</td>
<td></td>
</tr>
<tr>
<td>Callee tel-event payload type:</td>
<td>Default</td>
<td></td>
</tr>
<tr>
<td>Media flag:</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>Restrict codecs to list:</td>
<td>Default</td>
<td></td>
</tr>
<tr>
<td>Restrict caller codecs to list:</td>
<td>Default</td>
<td></td>
</tr>
<tr>
<td>Restrict callee codecs to list:</td>
<td>Default</td>
<td></td>
</tr>
<tr>
<td>Codec preference list:</td>
<td>Default</td>
<td></td>
</tr>
<tr>
<td>Caller Codec profile:</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>Callee Codec profile:</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>Caller media caps list:</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>Callee media caps list:</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>TCS extra codec list:</td>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>
Configuration Example of BFCP Support

Following is a configuration example of the BFCP Support feature on the SBC:

```
sbc sbc
  sbe
    stream-list my-stream
description voip stream list
generic-stream media-type application transport udp protocol BFCP
generic-stream media-type application transport udp protocol test
  exit
  cac-policy-set 2
  cac-table 2
table-type limit src-adjacency
  entry 1
    action cac-complete
generic-stream caller my-stream
generic-stream callee my-stream
    match-value SIP-adj-test
  exit
  exit
  complete
```
Chapter 32 BFCP Support

Configuration Example of BFCP Support
H.323 Support

Cisco Unified Border Element (SP Edition) supports H.323, as well as Session Initiation Protocol (SIP). This H.323 interworking capability enables multimedia products and applications from multiple vendors to interoperate and allows users to communicate without concern for compatibility.

H.323 is the international standard for multimedia communication over packet-switched networks, including local area networks (LANs), wide area networks (WANs), and the Internet. It was first defined by the International Communications Union (ITU) in 1996. The most recent version is H.323 version 6 (2006).

Note

This feature is supported in the unified model for Cisco IOS XE Release 2.5 and later.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).

For a complete description of the commands used in this chapter, refer to the Cisco Unified Border Element (SP Edition) Command Reference: Unified Model at:


For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

Feature History for H.323 Support

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 2.5</td>
<td>H.323 feature support was introduced on the Cisco ASR 1000 Series Aggregation Services Routers.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 3.2S</td>
<td>Added the restrictions for the support of SIP secure calls over an H.323 interface.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 3.3S</td>
<td>The Limited H.323 ID Routing and Passthrough Support feature was added.</td>
</tr>
</tbody>
</table>
Prerequisites for H.323 Support

This feature requires basic understanding of H.323-related ITU standards, gatekeepers, and gateways. Gateways are responsible for edge routing decisions between the Public Switched Telephone Network (PSTN) and the H.323 network. Gatekeepers are used to group gateways into logical zones and perform call routing between them.

Restrictions for H.323 Support

The restrictions for H.323 support are listed per feature in this chapter and other H.323-related chapters in the guide.

Information About H.323 Support

H.323 is a suite of protocols and documents that includes the ITU-T standards H.323, H.225.0, H.245, the H.450-series, and the H.460-series. Not all components of H.323 are mandatory as part of a standard H.323 system. For example, H.460.2, which describes number portability, is generally not used in enterprise video conferencing systems. Also H.323 utilizes both ITU-defined codecs and codecs defined outside the ITU to transmit audio, video, and text.

H.323 is used in Voice over Internet Protocol (VoIP), and IP-based video conferencing and serves a similar purpose to that of the Session Initiation Protocol (SIP). It was designed from the outset to operate over IP networks primarily, although H.323 may also operate over other packet-switched networks. H.323 was designed with multipoint voice and video conferencing capabilities, although most users do not take advantage of the multipoint capabilities specified in the protocol.

H.323 is more mature than SIP, but lacks the flexibility of SIP. SIP is currently less defined, but has greater scalability which could ease the Internet application integration. Like SIP, H.323 is one of the world market leaders for transporting voice and video over networks around the world, with billions of minutes of voice traffic every month. The SBC supports both SIP and H.323, enabling multimedia products and applications from multiple vendors to interoperate, and allowing users to communicate without concern for compatibility.

The following supported H.323 features are documented in another chapter in this configuration guide or represent part of standard Q.931/H.225 protocols that are documented in this chapter:
• H.323 to SIP Interworking—Interworking of a defined subset of SIP/H.323 call and media signaling. See the H.323 to SIP Interworking chapter in this configuration guide.

• Basic conferencing passthrough (this feature is part of Q.931/H.225 passthrough)—Pass through of conferenceID and conferenceGoal. Conference is controlled by the third party equipment, such as call manager. The SBC enables the conference to pass through all the conference-related information.

• H.450 passthrough (this feature is part of Q.931/H.225 passthrough)—Pass through of H.450 elements between call legs.

**Note**
All H.323 calls, including established H.323-H.323 and SIP-H.323 interworking calls, are disconnected upon an SBC switchover. An SBC switchover occurs when an active RP switches over to the standby RP in a hardware redundant system (such as a Cisco ASR 1006 Router) or when the active IOS process switches over to the standby IOS process in a redundant software system (such as a Cisco ASR 1002 Router).

---

### H.323 Features

The following supported H.323 features are documented in this chapter. The H.323 Call Routing features are documented in the Chapter [CT ChapTitle] chapter:

- H.323 Call Routing
- H.323 Video Codec Support
- H.323 Slow Start to H.323 Fast Start Interop
- Separate H.245 Control Channel
- H.245 Passthrough
- Slow Start Media Relay
- Codec Mappings
- DTMF Interworking
- Transcoding
- RAS Tech Prefix
- User Protocol Timer Control
- T.38 Fax Relay
- Q.931/H.225 Passthrough
- H.323 Privacy
- H.245 Address in Call Proceeding
- Multiple TCP for H.323
- Extending SIP Secure calls over H.323 Interface
- Limited H.323 ID Routing and Passthrough Support
H.323 Call Routing

Cisco Unified Border Element (SP Edition) supports the following H.323 call routing features:

- H.323 Hunting
- Picking a Next Hop in Routing Policy
- Support for H.323 Addressing
- DNS Name Resolution
- Number Validation and Editing
- Load Balancing
- Inter-VPN Calling

Note: The H.323 call routing features are noted in this chapter for reference. However, they are described in the H.323 Call Routing Features in the Chapter 33 Call Routing Features chapter.

H.323 Video Codec Support

Cisco Unified Border Element (SP Edition) allows H.323 video calls to be established through it. The supported H.323 video codecs are H.261, H.263 and H.264.

No specific configuration is required on either the end points or the SBC to enable this feature.

H.323 Slow Start to H.323 Fast Start Interop

Cisco Unified Border Element (SP Edition) supports the ability of H.323 endpoints with different starting modes of operation to interoperate with one another. Note that only the H.323 slow start endpoint to H.323 fast start endpoint is supported, and not vice-versa.

H.323 has two modes of operation: slow start and fast start. The initiation of a call may proceed in a slow start or fast start in H.323. In a slow start, H.323 signaling consists of Setup, Call Proceeding, Alerting, and Connect steps. After these steps, the H.245 media negotiation is performed. When a call is initiated in H.323 fast start, the H.245 media negotiation is performed within the initial Setup message.

This H.323 Slow Start to H.323 Fast Start Interop feature is enabled on a per adjacency basis. You can use the start fast command to configure the H.323 fast start mode. All calls routed to an adjacency uses the fast start mode that is configured on that adjacency. H.323 endpoints start in the fast start mode for outgoing calls on the adjacency and be able to convert incoming calls to the mode configured for that endpoint on the adjacency. When the fast start mode is configured, the SBC only uses the fast start mode for outgoing calls. However, incoming slow start calls are converted to fast start mode as they cross the SBC.

If the fast start mode is not configured on the adjacency, by default, the outgoing call start is the same as the incoming call start. The mode of operation can be modified while the adjacency is active but the change will only affect new calls. See the Chapter 33 Configuring H.323 Slow Start to H.323 Fast Start Interop section on page 33-21 for configuration step information.
Note

The fast start outgoing call is only a proposal, indicating the preferred mode from the SBC’s perspective. The H.323 endpoint can accept it or fall back to normal slow start procedure according to the H.245 specification.

Separate H.245 Control Channel

The H.323 procedures require that the SBC sets up a separate H.245 control channel over TCP. This feature complements tunneled H.245 support, enabling the user to control whether to use tunneling or not.

This feature enables the SBC to carry out an H.323-H.323 call, where two call legs can negotiate different H.245 transport mechanisms. Each call leg decides independently whether to use a separate H.245 control channel.

The SBC sets up separate H.245 control channels only when required in one of the following cases:

- The SBC has received a startH245 Facility
- The SBC needs to send out an H.245 message and tunneling is not available

The SBC never requests separate H.245 control channel while tunneling is available unless the "disable tunneling" command line interface (CLI) command is set (see $paranum>Configuring Separate H.245 Control Channel? section on page 33-22). The SBC does not connect to an H.245 address simply because the peer offered an h245Address.

The SBC does not offer an H.245 address until it needs to, performing the following:

- Where possible, the SBC connects to the peer instead.
- Where impossible, the SBC offers its own H.245 address in a startH245 Facility and waits for 10 seconds for the peer to connect. This timeout is not configurable.

Since H.323 v2 onwards has support for Facility reason startH245, support for this feature is assumed in all peer devices. If the peer requires an H.245 connection and one does not exist, the partner must use a startH245 Facility to induce the SBC to connect to it.

If there is no H.245 transport possible (tunneled or separate), and H.245 messages must be sent by the SBC, then the call is terminated.

On receipt of provisionalRespToH245Tunnelling, the SBC waits to determine the final tunneling outcome before attempting separate H.245. H.245 messages are queued at this point and sent as soon as an H.245 transport becomes available.

In the event of an H.245 connection race, the SBC only disconnects if it looses. The partner must disconnect if it loses. Races are resolved by comparing the listen address/port (not the connection address/port).

Back-pressure is exerted at call scope, or connection scope when multiple calls share a connection. So, if call leg B cannot forward H.245 messages for some reason, call leg A’s connection may exert TCP back pressure on the peer. If call leg A is doing H.245 tunneling, and sharing a Q.931 TCP connection with other calls, then the peer will experience back pressure on the other calls too.

The SBC tears down separate H.245 connections at the same point as their call by closing the relevant socket.
Restrictions for Separate H.245 Control Channel

The restrictions for the H.245 control channel are:

- The SBC does not support a model, in which it induces the peers in an H.323-H.323 call to set up the H.245 TCP connection directly between themselves or to the data border element (DBE).
- No show command is provided to list the H.245 transport status on a per-adjacency or per-call basis.
- The H.245 security is not supported.

H.245 Passthrough

In media bypass, H.245 content is passed unmodified between two H.323 call legs (for more information about media bypass, see the "How Adjacencies Affect Media Routing?" section on page 6-5 in the Implementing Adjacencies on Cisco Unified Border Element (SP Edition) chapter). Passthrough happens irrespective of whether an H.245 message is received over tunneled H.245 transport or a separate H.245 control channel, and does not require that both H.323 call legs use the same H.245 transport mechanism. This feature permits inserting an SBC between two H.323 devices without any change to the passing H.245 content.

This is achieved by passing through H.245 messages opaquely between the endpoints. The Fast Start request and response is passed through in the same way as mainline H.245. The only messages inspected by the SBC are fast start and logical channel signaling. These are used to derive the bandwidth used for the call.

Restrictions for H.245 Passthrough

The restrictions for the H.245 passthrough are:

- Configuration to block passthrough of certain messages or message elements is not included in this feature and is covered separately.
- In a media bypass call, no Session Description Protocol (SDP) appears in the billing records.
- The SBC does not support rate limiting of passed-through H.245 traffic, other than generic rate limiting of all signaling traffic.

Slow Start Media Relay

The SBC supports media relay (which is media pass through the DBE) of unidirectional H.245 channels. H.245 codec types are converted to Session Description Protocol (SDP) for the purposes of the DBE programming, transcoder programming, and billing. This is done using a codec mapping table (see "T.38 Fax Relay? section on page 33-11").

When dealing with codec types, for which no SDP mapping exists, the SBE makes a best-effort attempt, and tries to find the best possible SDP match. You can also use the `codec-restrict-to-list` command to configure a Call Admission Control (CAC) policy to restrict the codecs used in signaling a call to the set of codecs given in the named list. This configured CAC policy will have the effect of blocking setup of a particular codec or of ignoring an unknown codec.

Inserting an SBC between two H.323 devices does not impact H.245 function (see "H.245 Passthrough? section on page 33-6"). For example, the SBC does not modify the logical channel numbers of H.245 channels in a media relay call. In a distributed DBE model, H.248 signals are used to establish the necessary media terminations on the DBE.
The SBC supports renegotiation of media, using H.245 procedures, such as:

- Fax upspeed: Where endpoints switch over from a low-bit-rate audio codec to ITU-T G.711
- TCS=0: Where one endpoint induces the other to temporarily close all of its channels

Switchover to T.38 fax is described below in T.38 Fax Relay section on page 33-11.

**Restrictions for Slow Start Media Relay**

The restrictions for slow start media relay are:

- The SBC does not support bidirectional H.245 channels in fast start or Open Logical Channel (OLCs).
- Dual tone multifrequency (DTMF) interworking is not supported between different types of UserInputIndication.
- For SIP-SIP and H.323-H.323 calls, no user configuration is needed for DTMF interworking, which is triggered solely by capability negotiation.
- The SBC does not support multipoint capabilities.

**Codec Mappings**

The following codec mappings (Table 20-1) are used by the SBC to represent H.245 codecs as SDP for the purpose of:

- Billing records (media relay only)
- DBE programming (media relay only)
- Bandwidth allocation (media relay and media bypass). The bandwidth here is calculated based on the SDP, not directly from the H.245.

<table>
<thead>
<tr>
<th>H.245 codec</th>
<th>appears as:</th>
</tr>
</thead>
<tbody>
<tr>
<td>g711A-law 64k</td>
<td>PCMA/8000</td>
</tr>
<tr>
<td>g711U-law 64k</td>
<td>PCMU/8000</td>
</tr>
<tr>
<td>g722_64k</td>
<td>G722/8000</td>
</tr>
<tr>
<td>g723i</td>
<td>G723.1/8000</td>
</tr>
<tr>
<td>g728</td>
<td>G728/8000</td>
</tr>
<tr>
<td>g729</td>
<td>G729/8000</td>
</tr>
<tr>
<td>g729A</td>
<td>G729/8000</td>
</tr>
<tr>
<td>g729wA</td>
<td>G729/8000</td>
</tr>
<tr>
<td>g729wAAnnexB</td>
<td>G729/8000</td>
</tr>
<tr>
<td>gsmHalfRate</td>
<td>GSM-HR/8000</td>
</tr>
<tr>
<td>gsmFullRate</td>
<td>GSM/8000</td>
</tr>
<tr>
<td>All other audio codecs</td>
<td>PCMU/8000 (the default codec)</td>
</tr>
<tr>
<td>T.38</td>
<td>See T.38 Fax Relay, page 33-11</td>
</tr>
</tbody>
</table>

Note the following:
H.323 Features

- H.245 video and data codecs, other than T.38 and also H.261, H.263, H.264 for H.323-H.323 calls, are not supported by the SBC for media relay and media bypass.
- The subset of codecs supported for H.323/SIP interworking is much smaller (for more information see the H.323 to SIP Interworking chapter)

For general information, see the Codec Handling chapter.

DTMF Interworking

Dual-tone multi-frequency (DTMF) tones are used to transfer user requests. Different systems may support different forms of DTMF. The SBC enables the DTMF interworking between these systems.

For example, some nonstandard H.323 devices do not support the lowest common denominator of alphanumeric UserInputIndication. Such devices can only signal DTMF through RFC2833 telephony events or as in-band media data. Other devices support UserInputIndication but not the RFC2833 telephony event.

If two such devices are deployed back to back, their only option is to send DTMF tones as it is done in-band media data. Deploying an SBC between them allows each side to send DTMF, using its supported signaled method, UserInputIndication on one side and RFC2833 on the other, with the SBC interworking between the two.

This function requires the SBE to program the DBE to enable interception and insertion of RFC2833 DTMF on a particular side of the call—the side facing the RFC2833-only device. The SBE and DBE then collaborate to transfer DTMF signaling between the H.245 control channel and the RTP stream.

DTMF interworking is negotiated through TerminalCapabilitySet. Therefore, the SBC must be capable of extending the TerminalCapabilitySet to advertise support for both alphanumeric and RFC2833 methods.

The feature described in this section replaces all previous H.323 DTMF interworking functions. H.323 calls must support DTMF interworking between alphanumeric UserInputIndication and RFC2833. In this case, the SBE coordinates with the DBE to carry out DTMF insertion and interception in the Real-Time Protocol (RTP).

DTMF interworking is negotiated through TerminalCapabilitySet, not manual configuration. Therefore, the SBC must always advertise support for both alphanumeric and RFC2833 methods, if necessary by extending the TerminalCapabilitySet on its way through. (The exception is a TCS=0.)

See also the Implementing Interworking DTMF chapter for general information.

Restrictions for DTMF Interworking

The restrictions for DTMF internetworking are as follows:

- The alphanumeric UserInputIndication method and DTMF RFC 2833 are supported for DTMF interworking.
- The SBE assumes that a peer, advertising any form of UserInputCapability is capable of sending and receiving alphanumeric DTMF.
- No manual configuration is provided to force DTMF interworking to occur.
- Detection or insertion of DTMF as in-band audio data is not supported.
Transcoding

The SBC supports transcoding of slow start calls, enabling communication between different endpoints with different codecs, which otherwise cannot communicate with each other. Two H.323 endpoints deployed back-to-back might fail to agree on a mutually acceptable codec.

A typical case might be where one side is insisting on a low-bandwidth codec (such as ITU-T G.729) because of bandwidth constraints or administrative policy, and the other side only supports G.711. For example, if the calling party uses g711alaw and the callee uses G.729 annex B, the SBC can convert G.711alaw codec to g729 annex B codec and enable communication between the two. When the SBC detects that codec negotiation is needed, it uses Cisco Voice Interworking Service Module (VXSM) in the Cisco MGX 8880 switch as its media gateway to perform the transcoding. Deploying the SBC between the endpoints, in conjunction with an MGX 8880 transcoder, allows such calls to succeed.

The previous releases supported a fast-start-only version of transcoding. This function is now replaced with an implementation of transcoding that is triggered off TerminalCapabilitySet.

Transcoding is supported only for SIP-SIP calls.

See also the chapter for general transcoding information.

Restrictions for Transcoding

The restrictions for H.323 transcoding are:

- No transcoding support for SIP-H.323 and H.323-H.323 calls.
- The decision whether to use a transcoder is taken once per call, and is not modified if endpoints issue updated TerminalCapabilitySets (including TCS=0).
- When transcoding is required, the SBC enforces symmetric codecs for the call.
- Transcoding is never invoked in a fast start call. If no channels are suitable, endpoints must drop to slow start at which point transcoding may be invoked.
- The only codecs supported for transcoding are G.711 (PCMU and PCMA) and G.729 (with and without annex B), and the only transcoder tested with them is the Cisco MGX 8880 switch.

RAS Tech Prefix

A technology prefix is an optional H.323 standard-based feature, supported by gateways and gatekeepers, that enables more flexibility in call routing within an H.323 VoIP network. The gatekeeper uses technology prefixes to group endpoints of the same type together. Technology prefixes can also be used to identify a type, class, or pool of gateways. This feature provides per-adjacency configuration of RAS Tech Prefix and registers this prefix with the gatekeeper.

An H.323 adjacency may now be optionally configured with a single tech prefix consisting of 1-32 dialed digits. It publishes the tech prefix to the gatekeeper in the following field of the RAS registration request (RRQ):

terminalType.gateway.protocol.voice.supportedPrefixes.

As with existing adjacency configuration, this field may not be changed while the adjacency is attached. This feature works in conjunction with existing SBC support for adding or removing digits on dialed numbers (see the Number Manipulation section in the chapter).
Restrictions for RAS Tech PREFIX

- This feature does not support zone prefixes, for example, registration of prefix Alias Addresses with the gatekeeper.

User Protocol Timer Control

H.323 standards recommend timers, timeout, and retry counts for various messages. Their values are not fixed and represent a range. The ability to define these values facilitates interworking between different devices. H.323 timers and retry counts can be now configured by the user at a global and per-adjacency level. Timers are expressed in seconds.

The following Q.931/H.225 timers are configurable.
- Q.931/H.225 Setup Timer T303
- Q.931/H.225 Establishment Timer T301
- Q.931/H.225 Incoming Call Proceeding Timer T310

The following RAS timeout and retry counts are configurable.
- GRQ
- RRQ
- URQ
- ARQ
- BRQ
- DRQ

The RAS RRQ TTL and keepalive times (governing lightweight RRQ behavior) are configurable. These two settings are interrelated. If the user configures unsafe values for a given adjacency, the SBE reverts to the defaults.

The adjacency retry timer is configurable, and can be used to automatically reattempt adjacency attachment when an adjacency fails for any reason.

The following timers are hardcoded:
- TCP shutdown timeout—when gracefully closing a TCP connection, the time allowed for remote closure before closing it ungracefully. The hardcoded value is 1 second.
- TCP connect timeout—time allowed before giving up on a TCP connection attempt to a remote peer. The hardcoded value is 1 second.

Restrictions for User Protocol Timer Control

User protocol timer control restrictions are:
- Changing timer values or retry counts while adjacencies are attached is allowed, but does not affect timers’ or gatekeeper’s transactions that are already in progress.
- No facility is provided to configure all RAS timeouts at once.
- H.245 timers are not included here since they only run in interworking scenarios.
- The SBC does not support the configuration of the following Q.931/H.225 timers:
  - Q.931/H.225 Overlap Sending Timer T302
Chapter 33  H.323 Support

H.323 Features

- Q.931/H.225 Overlap Receiving Timer T304
- Q.931/H.225 Status Timer T322

- The SBC does not support the configuration of the following RAS timers:
  - IRQ
  - IRR
  - RAI
  - SCI

T.38 Fax Relay

This feature provides support for media relay of T.38 fax. The following features are supported:

- Both fax-only and fax-plus voice calls.
- Switchover from voice to T.38 fax.
- T.38 relay over unnumbered datagram protocol transport layer (UDPTL) only, and unidirectional H.245 channels only.

T.38 H.245 - SDP Mapping

The T.38 H.245—SDP mapping is shown below:

```
data:ApplicationCapability
  application: t38fax
  t38FaxProtocol
    m=image 40000 (udptl | tcp) t38
  t38FaxProfile
    fillBitRemoval a=T38FaxFillBitRemoval
    transcodingJBIG a=T38FaxTranscodingJBIG
    transcodingMMR a=T38FaxTranscodingMMR
    version a=T38FaxVersion:<digits>
    t38FaxRateManagement a=T38FaxRateManagement:(localTCF | transferredTCF)
    t38FaxUdpOptions OPTIONAL
      t38FaxMaxBuffer a=T38FaxMaxBuffer:<digits>
      t38FaxMaxDatagram a=T38FaxMaxDatagram:<digits>
      t38FaxUdpEc a=T38FaxUdpEc:{t38UDPEnc | t38UDPRedundancy}
    t38FaxTcpOptions OPTIONAL
      t38TCPBidirectionalMode [no mapping]
  maxBitRate a=T38maxBitRate:<digits> (UDP only)
```

The only parameters needed for media relay function are the port and the peak-bit rate, which are highlighted in the example. Therefore, the presence of a T.38 fax function causes the following SDP to be transmitted to the DBE:

```
m=image <remote T.38 port> udptl t38
a=T38maxBitRate:14400
```

For interworking scenarios, a complete mapping needs to be carried out.
H.245 Mode Request

Switchover from a voice to fax call is handled by a RequestMode exchange. In an H.323-H.323 call this exchange is passed through transparently between call legs without DBE signaling. This allows endpoints to coordinate replacement of audio with T.38 channels.

RAS Maximum Bit Rate

In accordance with H.323v5 standards, the SBC counts UDP but not TCP towards the maximum bit rate agreed with the gatekeeper.

H.323 Annex D / T.38 Annex B Interoperability

T.38 Annex B is a fast start only (no H.245) version of H.323 Annex D. Interoperation with Annex B nodes is not supported by the SBC.

Restrictions

The restrictions are as follows:

- The SBC cannot be configured to advertise the t38FaxAnnexbOnly field of SupportedProtocols in RAS messages, and ignores this field on receipt.
- No support for TCP or Secure Real-Time Transport Protocol (SRTP) transport.
- No support for bidirectional H.245 channel signaling.

Q.931/H.225 Passthrough

This feature enables message elements from Q.931/H.225 to be passed through between two H.323 call legs. This section describes the "base passthrough profile" of the SBC, listing the parts of the Q.931/H.225 message that may be passed through.

The following conventions are used in the base passthrough profile:

- ASN.1 syntax for Q.931 / H.225 messages is reproduced in this document.
- The following tags are attached to ASN.1 subtrees, specifying the passthrough behavior.
  - P = "passthrough". This subtree is passed opaquely between call legs.
  - P* = "passthrough with privacy implications". Similar to "P", but passing through this subtree may reveal information about an endpoint or a remote telephone number.
  - B = "block". This subtree is unconditionally blocked by the SBC and any information contained in it is lost.
  - SBC. This subtree is manipulated by the SBC. Typically, values are replaced by those local to the SBC.

Call Proceeding Passthrough

A Call Proceeding message is never passed through. However, fields from it are extracted and put into a Progress or Facility in the upstream call log.

- A Progress is used if the Call Proceeding contains a progress indicator.
• A Facility is used otherwise.

Unsupported Messages

The following ITU-T Q.931 messages are not supported by the SBC either because they are forbidden in H.323 or because the SBC does not currently support their corresponding features.

• Status, Status Enquiry
• SetupAck
• Information
• Notify
• userInformation.

Privacy

Subtrees marked as "P*" - "passthrough with privacy implications" are automatically blocked if the outgoing call leg has privacy enabled in CAC policy. This automatic blocking cannot be overridden by configuration, therefore, the only way to have these fields pass through is to disable privacy.

Setting of Protocol Version

When passing through messages, the SBC sets the version of outgoing messages to the lower value of its own ASN.1 version from that received in the original protocol message.

Q.931 / H.225 Base Passthrough Profile

```plaintext
Q931Message
  protocolDiscriminator SBC
  callReferenceValue SBC
  message
    setup
      sendingComplete P
      bearerCapability P
      facility P
      progressIndicator P
      progressIndicator31 P
      notificationIndicator P
      display P*
      keypadFacility P
      signal P
      callingPartyNumber SBC
      callingPartySubaddress B
      calledPartyNumber SBC
      calledPartySubaddress B
      redirectingNumber P*
    userUser
    h323-uu-pdu
    h323-message-body
      setup
        protocolIdentifier SBC
        h245Address SBC
        sourceAddress SBC
        sourceInfo SBC
        destinationAddress SBC
        destCallSignalAddress SBC
```
destExtraCallInfo             B
destExtraCRV                  B
activeMC                      P
conferenceID                  P
conferenceGoal                P
callServices                  P
callType                      B
sourceCallSignalAddress       SBC
remoteExtensionAddress        B
callIdentifier                P
h245SecurityCapability        B
tokens                        B
cryptoTokens                  B
fastStart                      SBC
mediaWaitForConnect           P
canOverlapSend                B
endpointIdentifier            P*
multipleCalls                 SBC
maintainConnection            SBC
connectionParameters          P
language                      P
presentationIndicator         SBC
screeningIndicator            SBC
serviceControl                P
symmetricOperationRequired    P
capacity                      B
circuitInfo                   SBC
desiredProtocols              B
neededFeatures                 B
desiredFeatures               B
supportedFeatures             B
parallelH245Control           B
additionalSourceAddresses     B
nonStandardData               P
h4501SupplementaryService     P
h245Tunneling                 SBC
h245Control                   SBC
nonStandardControl            P
callLinkage                   P
tunnelledSignallingMessage    P
provisionalRespToH245Tunneling SBC
stimulusControl               P
genericData                   P
user-data                      P
callProceeding                
bearerCapability              P
facility                      P
progressIndicator             SBC
progressIndicator31           SBC
notificationIndicator         P
display                       P
userUser                      
h323-uu-pdu
h323-message-body

callProceeding
 protocolIdentifier         SBC
 destinationInfo             P*
h245Address                  SBC
callIdentifier               P
h245SecurityMode             B
tokens                       B
cryptoTokens                 B
fastStart                     SBC
multipleCalls                SBC
maintainConnection             SBC
fastConnectRefused            SBC
featureSet                    B
nonStandardData               P
h4501SupplementaryService     P
h245Tunneling                 SBC
h245Control                  SBC
nonStandardControl            P
callLinkage                   P
tunnelledSignallingMessage    P
provisionalRespToH245Tunneling SBC
stimulusControl               P
genericData                   P
user-data                     P
alerting
bearerCapability             P
facility                     P
progressIndicator            SBC
progressIndicator31          SBC
notificationIndicator         P
display                      P*
signal                       P
userUser
h323-uu-pdu
h323-message-body
alerting                     P
protocolIdentifier            SBC
destinationInfo               P*
h245Address                  SBC
callIdentifier                P
h245SecurityMode              B
tokens                       B
cryptoTokens                  B
fastStart                     SBC
multipleCalls                 SBC
maintainConnection            SBC>alertingAddress               P*
presentationIndicator       SBC
screeningIndicator           SBC
fastConnectRefused           SBC
serviceControl               P
capacity                    B
featureSet                   B
nonStandardData              P
h4501SupplementaryService    P
h245Tunneling                SBC
h245Control                 SBC
nonStandardControl           P
callLinkage                  P
tunnelledSignallingMessage   P
provisionalRespToH245Tunneling SBC
stimulusControl              P
genericData                  P
user-data                    P
connect
bearerCapability             P
facility                     P
progressIndicator            SBC
progressIndicator31          SBC
notificationIndicator        P
display                      P*
dateTime                     P
connectedNumber              P*
connectedSubaddress          P*
H.323 Support

H.323 Features

userUser
h323-uu-pdu
h323-message-body
connect
protocolIdentifier SBC
h245Address SBC
destinationInfo P*
conferenceID P
callIdentifier P
h245SecurityMode B
tokens B
cryptoTokens B
fastStart SBC
multipleCalls SBC
maintainConnection SBC
language P
connectedAddress P*
presentationIndicator SBC
screeningIndicator SBC
fastConnectRefused SBC
serviceControl P
capacity B
defeatureSet B
nonStandardData P
h4501SupplementaryService P
h245Tunneling SBC
h245Control SBC
nonStandardControl P
callLinkage P
tunnelledSignallingMessage P
provisionalRespToH245Tunneling SBC
stimulusControl P
genericData P
user-data P
progress
bearerCapability P
cause P
callLinkage P
progressIndicator SBC
progressIndicator31 SBC
notificationIndicator P
display P*
userUser
h323-uu-pdu
h323-message-body
progress
protocolIdentifier SBC
destinationInfo SBC
h245Address SBC
callIdentifier P
h245SecurityMode B
tokens B
cryptoTokens B
fastStart SBC
multipleCalls SBC
maintainConnection SBC
fastConnectRefused SBC
nonStandardData P
h4501SupplementaryService P
h245Tunneling SBC
h245Control SBC
nonStandardControl P
callLinkage P
tunnelledSignallingMessage P
H.323 Support

H.323 Features

provisionalRespToH245Tunneling  SBC
stimulusControl                  P
genericData                      P
user-data                        P
releaseComplete                  SBC
cause                            P
facility                         P
notificationIndicator            P
display                          P*
signal                           P
userUser
h323-uu-pdu
h323-message-body
  connect
    protocolIdentifier             SBC
    reason                         SBC
    callIdentifier                 P
    tokens                         B
    cryptoTokens                   B
    busyAddress                    P*
    presentationIndicator          SBC
    screeningIndicator              P
    capacity                       B
    serviceControl                 P
    featureSet                     B
    nonStandardData                P
    h4501SupplementaryService      P
    h245Tunneling                   SBC
    h245Control                    SBC
    nonStandardControl             P
    callLinkage                    P
    tunnelledSignallingMessage     P
    provisionalRespToH245Tunneling  SBC
    stimulusControl                P
    genericData                    P
    user-data                      P
    facility
    facility
    notificationIndicator          P
    display                        P*
    callingPartyNumber             P*
    calledPartyNumber              P*
    userUser
h323-uu-pdu
h323-message-body
  facility
    protocolIdentifier             SBC
    alternativeAddress             B
    alternativeAliasAddress        P
    conferenceID                   P
    reason                         P
    callIdentifier                 P
    destExtraCallInfo              P
    remoteExtensionAddress         P
    tokens                         B
    cryptoTokens                   B
    conferences                    P
    h245Address                    SBC
    fastStart                      SBC
    multipleCalls                  SBC
    maintainConnection             SBC
    fastConnectRefused             SBC
    serviceControl                 P
    circuitInfo                    B
Restrictions

- Any message elements from Q.931/H.225 that are not listed in this section cannot be passed through.
- Passsthrough of security tokens is not supported.

H.323 Privacy

With the H.323 privacy feature, users can invoke identity hiding on Q.931/H.225 messages. When this feature is implemented, the SBC strips Q.931/H.225 message elements that reveal information about the remote caller or callee before passing them to the endpoints.

The Q.931/H.225 message elements that impact privacy are defined in the H.323 passthrough profile.

The SBC applies the privacy service to a message if it contains a privacy request submitted by a user, or if a Call Admission Control (CAC) policy on the SBC is configured to enable privacy on a caller or callee basis. If, however, the privacy configuration fields are set to default values, then the SBC forwards the message to the next call leg without applying the privacy service to the message. You can also configure the SBC to provide the H.323 privacy service on a per-adjacency basis.

The SBC applies the following rules when providing the H.323 privacy service:

- If an H.323 adjacency is configured to allow private information, then the SBC does not apply privacy service even if an incoming message requests it or the CAC policy is configured to enable privacy.
- If an H.323 adjacency is not configured to allow private information, but a CAC policy is configured to enable privacy, then the SBC applies the privacy service to outgoing messages.
- If an incoming message requests the privacy service, but a CAC policy has not been configured to enable privacy, then the SBC applies the service if the adjacency is configured to apply the privacy service.
- If an incoming message requests the privacy service when both the CAC policy and the adjacency have not been configured to apply the privacy service, then the SBC does not apply the privacy service and allows the private information to pass through.

Restrictions and Limitations

Restrictions and limitations are as follows:
The SBC does not apply the H.323 privacy service to H.245 and RAS messages.

Currently, the CAC policy for callee privacy is available for the H.323 signaling stack at “connect time”, and only if a connectedNumber is present. As a result, the callee privacy service is not applied to the Q.931 protocol messages that pass through before or after a call is connected when a connectedNumber is not present. Due to this limitation, the SBC forwards the Q.931 Alerting, Q.931 Progress, and Q.931 Release Complete messages without applying the privacy service request to them.

In an interworking call, the SBC only applies privacy requests based on the CAC policy.

**H.245 Address in Call Proceeding**

The SBC allows the address of the H.245 listening socket to be published in the Q.931 call proceeding message. When the caller does not support tunneling and an H.245 address published to the caller set to “wait-connect”, H.323 supplies only the H.245 address on the Q.931 connect. In default behavior, H.323 supplies the H.245 address on a Q.931 call proceeding, and all subsequent messages to the caller until the H.245 connection is opened.

**Multiple TCP for H.323**

The SBC supports multiple TCP connections for an H.323 call, such as one H.225 signaling channel and one H.245 signaling channel if required.

**Extending SIP Secure calls over H.323 Interface**

The Cisco IOS XE Release 3.2S extends the security feature, by extending support of secure calls coming from an H323 adjacency or from a SIP adjacency. Before this enhancement, SBC only supported SIP secure calls and SIP secure calls were not able to interwork with H.323 networks. After this enhancement, SIP secure calls received from SIP adjacency and routed over H323 adjacency can be sent by configuring the H323 adjacency as trusted. Also, calls coming from an H323 adjacency and if it is required to be treated as a secure call then you can configure the H323 adjacency as secured.

Following are the restrictions of the SIP Secure calls over an H.323 interface:

- The SBC does not signal secure H.323 calls using the procedure described in H.235. It also does not recognize the secure nature of the incoming H.323 calls using the H.235 procedures.
- The SBC does not use a TLS or IPSec to send call signalling for secure H.323 calls.

**Limited H.323 ID Routing and Passthrough Support**

The Tandberg T3 device is a high-end 3-screen video-conferencing product that uses H.323 signaling. To direct the 3 video streams, it routes calls using the H.323 ID. The Destination Number.Screen Identifier H.323 ID format is used. For example, 12221120008.left, indicates that the T3 device is identified using the e164 number 12221120008, and the video stream that is being negotiated is to be presented on the left side of the screen. Similarly, H.323 negotiations for other streams with .right and .centre suffixes, and the same numeric prefix can be made.
Prior to Cisco IOS XE Release 3.3S, these calls would fail when the SBC is inserted into the signaling path for the following reasons:

- The call is not routed correctly, because the SBC does not read the information in the H.225 destination address field, and this is the only field in which routing information is made available by the T3.
- The destinationAddress field is blocked by the SBC, so even if the call were routed correctly, the receiving T3 does not have the information of the screen that has to be presented with the video stream.

From Cisco IOS XE Release 3.3S, when the SBC receives a Q.931 message that contains no routing information in the standard called party number (CdPN) field, it examines the H.225 destinationAddress field. If the field contains information of type e164, the contents is copied in to the CdPN field of the message. However, if the field contains information of H.323 ID type, the SBC copies the numeric prefix of this field into the CdPN field of the message.


### Restrictions for Limited H.323 ID Routing and Passthrough Support

The Limited H.323 ID Routing and Passthrough Support feature has the following restrictions:

- This feature is not supported on an inbound SIP adjacency, as the configuration is done on the inbound H.323 adjacencies.
- If the inbound adjacency for a call is H.323, you can derive the caller and the callee numbers from the H.225 and H.323 ID addresses by setting `h225 address usage to prefix`. You cannot pass through the complete H323-ID format addresses directly.
- H.225 addresses cannot pass through during the H.323-SIP interworking because an AliasAddress field cannot be inserted in a SIP message.

### Configuring H.323 Features

This section contains the following:

- Configuring H.323 Slow Start to H.323 Fast Start Interop, page 33-21
- Configuring Separate H.245 Control Channel, page 33-22
- Configuring RAS Tech Prefix, page 33-23
- Configuring User Protocol Timer Control, page 33-24
- Configuring H.323 Privacy, page 33-26
- Configuring H.245 Address in Call Proceeding, page 33-27
- Configuring Limited H.323 ID Routing and Passthrough Support, page 33-28
Configuring H.323 Slow Start to H.323 Fast Start Interop

The following example describes how to configure the SBC to use the fast start mode of operation for outgoing calls; and to effect the interop capability that enables the incoming slow start calls to be converted to fast start calls as they cross the SBC.

SUMMARY STEPS

1. configure terminal
2. sbc service-name
3. sbe
4. adjacency h323 adjacency-name
5. start fast
6. exit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure terminal</td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td>Example: Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 2 sbc service-name</td>
<td>Enters the mode of an SBC service. Use the service-name argument to define the name of the SBC.</td>
</tr>
<tr>
<td>Example: Router(config)# sbc mySBC</td>
<td></td>
</tr>
<tr>
<td>Step 3 sbe</td>
<td>Enters the mode of the SBE function of the SBC.</td>
</tr>
<tr>
<td>Example: Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td>Step 4 adjacency h323 adjacency-name</td>
<td>Enters the mode of an SBE H.323 adjacency. Use the adjacency-name argument to define the name of the H.323 adjacency.</td>
</tr>
<tr>
<td>Example: Router(config-sbc-sbe)# adjacency h323 2651XM-5</td>
<td></td>
</tr>
<tr>
<td>Step 5 start fast</td>
<td>Specifies that the SBC uses the fast start mode of operation for call setup for outgoing calls. The interop capability enables the incoming slow start calls to be converted to fast start calls as they cross the SBC.</td>
</tr>
<tr>
<td>Example: Router(config-sbc-sbe-adj-h323)# start fast</td>
<td></td>
</tr>
<tr>
<td>Step 6 exit</td>
<td>Exits the SBE H.323 adjacency mode to the SBE mode.</td>
</tr>
<tr>
<td>Example: Router(config-sbc-sbe-adj-h323)# exit</td>
<td></td>
</tr>
</tbody>
</table>
Configuring Separate H.245 Control Channel

This command disables tunneling on a per-adjacency basis, facilitating interoperability with existing devices that are confused by tunneling. The command controls both incoming and outgoing calls.

SUMMARY STEPS

1. configure terminal
2. sbc service-name
3. sbe
4. adjacency h323 adjacency-name
5. h245-tunnel disable
6. exit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure terminal</td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 2 sbc service-name</td>
<td>Enters the mode of an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# sbc mysbc</td>
<td>Use the service-name argument to define the name of the SBC.</td>
</tr>
<tr>
<td>Step 3 sbe</td>
<td>Enters the mode of the SBE function of the SBC.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td>Step 4 adjacency h323 adjacency-name</td>
<td>Enters the mode of an SBE H.323 adjacency.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe)# adjacency h323 2651XM-5</td>
<td>Use the adjacency-name argument to define the name of the H.323 adjacency.</td>
</tr>
<tr>
<td>Step 5 h245-tunnel disable</td>
<td>Disables tunneling on a per-adjacency basis, facilitating interoperability with existing devices that are confused by tunneling. The command controls both incoming and outgoing calls. The default is tunneling enabled.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-adj-h323)# h245-tunnel disable</td>
<td></td>
</tr>
<tr>
<td>Step 6 exit</td>
<td>Exits the SBE H.323 adjacency mode to the SBE mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-adj-h323)# exit</td>
<td></td>
</tr>
</tbody>
</table>
Configuring RAS Tech Prefix

This feature provides per-adjacency configuration of RAS Tech Prefix and registers this prefix with the gatekeeper. RAS tech prefix may consist of 1-32 dialed digits.

SUMMARY STEPS

1. configure terminal
2. sbc service-name
3. sbe
4. adjacency h323 adjacency-name
5. tech-prefix tech-prefix
6. exit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure terminal</td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 2 sbc service-name</td>
<td>Enters the mode of an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# sbc mysbc</td>
<td>Use the service-name argument to define the name of the SBC.</td>
</tr>
<tr>
<td>Step 3 sbe</td>
<td>Enters the mode of the signaling border element (SBE) function of the SBC.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td>Step 4 adjacency h323 adjacency-name</td>
<td>Enters the mode of an SBE H.323 adjacency.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe)# adjacency h323 2651XM-5</td>
<td>Use the adjacency-name argument to define the name of the H.323 adjacency.</td>
</tr>
<tr>
<td>Step 5 tech-prefix tech-prefix</td>
<td>Provides per-adjacency configuration of RAS Tech Prefix and registers this prefix with the gatekeeper. RAS tech prefix may consist of 1-32 dialed digits followed by a # sign. The default is no tech prefix.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-adj-h323)# tech-prefix 32#</td>
<td></td>
</tr>
<tr>
<td>Step 6 exit</td>
<td>Exits the SBE H.323 adjacency mode to the SBE mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-adj-h323)# exit</td>
<td></td>
</tr>
</tbody>
</table>
Configuring User Protocol Timer Control

**SUMMARY STEPS**

1. `configure terminal`
2. `sbc service-name`
3. `sbe`
4. `h323 | adjacency h323 adjacency-name`
5. `adjacency timeout value`
6. `h225 timeout`
7. `ras retry`
8. `ras rrq ttl value`
9. `ras rrq keepalive value`
10. `ras timeout`
11. `exit`
12. `show sbc sbc-name sbe h323 timers`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 <code>configure terminal</code></td>
<td>Enables the global configuration mode.</td>
</tr>
</tbody>
</table>
| **Example:** Router
  
  `Router# configure terminal` | |
| Step 2 `sbc service-name` | Enters the mode of an SBC service. Use the `service-name` argument to define the name of the SBC. |
| **Example:**
  
  `Router(config)# sbc mySBC` | |
| Step 3 `sbe` | Enters the mode of the SBE function of the SBC. |
| **Example:**
  
  `Router(config-sbc)# sbe` | |
| Step 4 `h323 | adjacency h323 adjacency-name` | Enters the mode of either all H.323 adjacencies or a specified H.323 adjacency. Use the `adjacency-name` argument to define the name of the H.323 adjacency. |
| **Example:**
  
  `Router(config-sbc-sbe)# adjacency h323 2651XM-5` | |
| Step 5 `adjacency timeout value` | Defines the time in milliseconds, during which in case of failure to connect, the SBC keeps trying to reconnect to the remote signaling peer and receive keep-alive messages from it. The value range is 10000—30000. |
| **Example:**
  
  `Router(config-sbc-sbe-adj-h323)# adjacency timeout 10000` | |
### Command or Action

| Step 6 | h225 timeout [establishment timeout-value | proceeding timeout-value | setup timeout-value |

**Example:**
```
Router(config-sbc-sbe-adj-h323)# h225 timeout establishment 250000
```

- **Defines the time for waiting to receive H.225 messages.**
  - **establishment timeout-value**—h225 establishment state timeout value in milliseconds. The default is 180000. The value range is 30000-300000.
  - **proceeding timeout-value**—h225 proceeding state timeout value in milliseconds. 10000. The value range is 1000-30000.
  - **setup timeout-value**—h225 setup timeout value in milliseconds. The default is 4000. The value range is 1000-30000.

| Step 7 | ras retry [arq | brq | drq | grq | rrq | urq] retry count |

**Example:**
```
Router(config-sbc-sbe-adj-h323)#
ras retry arq 2
ras retry brq 2
ras retry drq 2
ras retry rrq 2
ras retry urq 2
```

- **Defines the number of times the system tries to re-send RAS messages in case of failure to send the messages.**
  - **arq retry count**—Number of times to retry an ARQ transaction.
  - **brq retry count**—Number of times to retry a BRQ transaction.
  - **drq retry count**—Number of times to retry a DRQ transaction.
  - **grq retry count**—Number of times to retry a GRQ transaction.
  - **rrq retry count**—Number of times to retry an RRQ transaction.
  - **urq retry count**—Number of times to retry an URQ transaction.
  - The value range is 0-30.

| Step 8 | ras rrq ttl value |

**Example:**
```
Router(config-sbc-sbe-adj-h323)# ras rrq ttl 100
```

- **Defines the time to live messages (TTL) in seconds for registration request (RRQ).**
  - The default is 60. The value range is 16—300.

| Step 9 | ras rrq keepalive value |

**Example:**
```
Router(config-sbc-sbe-adj-h323)# ras rrq keepalive 100000
```

- **Defines the time in milliseconds for registration request (RRQ) keep-alive messages.**
  - The default is 45000. The value range is 15000—150000.
Configuring H.323 Features

### Configuring H.323 Privacy

This feature allows the SBC to apply the H.323 privacy service on outbound messages.

**SUMMARY STEPS**

1. `configure terminal`
2. `sbc service-name`
3. `sbe`
4. `adjacency h323 adjacency-name`
5. `allow private info`
6. `privacy restrict outbound`
7. `exit`

---

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 10</strong> ras timeout [arq</td>
<td>brq</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-adj-h323)# ras timeout arq 1000 ras timeout brq 1000 ras timeout drq 1000 ras timeout grq 1000 ras timeout rrq 1000 ras timeout urq 1000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 11</strong> exit</td>
<td>Exits the H.323 global or specified adjacency mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-adj-h323)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 12</strong> show sbc service-name sbe h323 timers</td>
<td>Displays the values of all H.323 timers.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# show sbc mysbc sbe h323 timers</td>
<td></td>
</tr>
</tbody>
</table>
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><code>configure terminal</code></td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Router# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td><code>sbc service-name</code></td>
<td>Enters the mode of an SBC service. Use the <code>service-name</code> argument to define the name of the SBC.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Router(config)# sbc mysbc</code></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td><code>sbe</code></td>
<td>Enters the mode of the SBE function of the SBC.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Router(config-sbc)# sbe</code></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td><code>adjacency h323 adjacency-name</code></td>
<td>Enters the mode of an SBE H.323 adjacency. Use the <code>adjacency-name</code> argument to define the name of the H.323 adjacency.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Router(config-sbc-sbe)# adjacency h323 2651XM-5</code></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td><code>allow private info</code></td>
<td>Configures the H.323 adjacency to allow private information on messages sent out by the adjacency even if the CAC policy is configured to apply privacy service or the user requests privacy service. The <code>no</code> version of this command configures the H.323 adjacency to stop allowing private information from being sent out by the adjacency.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Router(config-sbc-sbe-adj-h323)# allow private info</code></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td><code>privacy restrict outbound</code></td>
<td>Configures the H.323 adjacency to apply privacy restriction on outbound messages if the user requests the privacy service. The <code>no</code> version of this command configures the H.323 adjacency to allow private information messages sent out by the adjacency.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Router(config-sbc-sbe-adj-h323)# privacy restrict outbound</code></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td><code>exit</code></td>
<td>Exits an SBE H.323 global or specified adjacency mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Router(config-sbc-sbe-adj-h323)# exit</code></td>
<td></td>
</tr>
</tbody>
</table>

### Configuring H.245 Address in Call Proceeding

This feature allows the address of the H.323 listening socket to be published in the Q.931 call proceeding message.

### SUMMARY STEPS

1. `configure terminal`
2. `sbc service-name`
3. `sbe`
4. `adjacency h323 adjacency-name`
5. `h245-address-pass wait-connect`
6. `exit`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><code>configure terminal</code> Enables the global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router# <code>configure terminal</code></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><code>sbc service-name</code> Enters the mode of an SBC service. Use the <code>service-name</code> argument to define the name of the SBC.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config)# <code>sbc mysbc</code></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><code>sbe</code> Enters the mode of the SBE function of the SBC.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc)# <code>sbe</code></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td><code>adjacency h323 adjacency-name</code> Enters the mode of an SBE H.323 adjacency. Use the <code>adjacency-name</code> argument to define the name of the H.323 adjacency.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe)# <code>adjacency h323 2651XM-5</code></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td><code>h245-address-pass wait-connect</code> Configures the H.323 adjacency to allow delay passing the H.245 address to caller. If set to <code>wait-connect</code>, H.323 supplies only the H.245 address on the Q.931 connect. The no form of this command shows default behavior, where H.323 supplies the H.245 address on a Q.931 call proceeding, and all subsequent messages to the caller until the H.245 connection is opened.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-adj-h323)# <code>h245-address-pass wait-connect</code></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td><code>exit</code> Exits an SBE H.323 global or specified adjacency mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-adj-h323)# <code>exit</code></td>
</tr>
</tbody>
</table>

**Configuring Limited H.323 ID Routing and Passthrough Support**

This section shows how to configure the Limited H.323 ID Routing and Passthrough Support feature:

- Configuring H.225 Address Passthrough, page 33-29
- Configuring H.225 Address Usage, page 33-30
Configuring H.225 Address Passthrough

This task shows how to configure the SBC to block the sourceAddress and destinationAddress fields in H.225 messages received on the H.323 adjacency. When the SBC is configured to block, these fields are not sent out of the outbound H.323 adjacency.

SUMMARY STEPS

1. configure terminal
2. sbc sbc-name
3. sbe
4. adjacency h323 adjacency-name
5. h225 address block
6. end
7. show sbc sbc-name sbe adjacencies adjacency-name detail

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure terminal</td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td>Example: Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 2 sbc sbc-name</td>
<td>Enters the SBC service mode. Use the sbe-name argument to define the name of the SBC.</td>
</tr>
<tr>
<td>Example: Router(config)# sbc mysbc</td>
<td></td>
</tr>
<tr>
<td>Step 3 sbe</td>
<td>Enters the SBE function mode of the SBC.</td>
</tr>
<tr>
<td>Example: Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td>Step 4 adjacency h323 adjacency-name</td>
<td>Enters the SBE H.323 adjacency mode. Use the adjacency-name argument to define the name of the H.323 adjacency.</td>
</tr>
<tr>
<td>Example: Router(config-sbc-sbe)# adjacency h323 h323adj</td>
<td></td>
</tr>
<tr>
<td>Step 5 h225 address block</td>
<td>The sourceAddress and destinationAddress fields in the H.225 message that are received on the H.323 adjacency are not passed through. By default, the sourceAddress and destinationAddress fields are not blocked.</td>
</tr>
<tr>
<td>Example: Router(config-sbc-sbe-adj-h323)# h225 address block</td>
<td></td>
</tr>
</tbody>
</table>
**Configuring H.323 Features**

**Configuring H.323 Address Usage**

This task shows how to interpret H.225 sourceAddress and destinationAddress fields when the Q.931 callingPartyNumber or calledPartyNumber fields are not present. When callingPartyNumber is not provided in the Q.931 part of the message, the sourceAddress in the H.225 is checked. Similarly, when the calledPartyNumber field is not present, the destinationAddress is checked.

**SUMMARY STEPS**

1. configure terminal
2. sbc sbc-name
3. sbe
4. adjacency h323 adjacency-name
5. h225 address usage {e164 | h323id}
6. end
7. show sbc sbc-name sbe adjacencies adjacency-name detail

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure terminal</td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td>Example: Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 2 sbc sbc-name</td>
<td>Enters the SBC service mode. Use the sbc-name argument to define the name of the SBC.</td>
</tr>
<tr>
<td>Example: Router(config)# sbc mysbc</td>
<td></td>
</tr>
<tr>
<td>Step 3 sbe</td>
<td>Enters the SBE function mode of the SBC.</td>
</tr>
<tr>
<td>Example: Router(config-sbc)# sbe</td>
<td></td>
</tr>
</tbody>
</table>
Chapter 33  H.323 Support

Configuring H.323 Features

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 4  adjacency h323 adjacency-name</td>
<td>Enters the SBE H.323 adjacency mode. Use the <em>adjacency-name</em> argument to define the name of the H.323 adjacency.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe)# adjacency h323 h323adj</td>
<td></td>
</tr>
</tbody>
</table>

| Step 5  h225 address usage {e164 | h323id}                      | Specifies either of the following interpretation formats for the H.225 sourceAddress and destinationAddress fields in the adjacency when Q.931 callingPartyNumber or calledPartyNumber is not present: |
| Example:                           |                                                                         |
| Router(config-sbc-sbe-adj-h323)# h225 address usage e164               |                                                                         |

| Step 6  end                        | Exits the SBE H.323 adjacency mode and enters the Privileged Exec mode. |
| Example:                           |                                                                         |
| Router(config-sbc-sbe-adj-h323)# end|                                                                         |

| Step 7  show sbc sbc-name sbe adjacencies adjacency-name detail        | Lists the adjacencies configured in the SBE.                             |
| Example:                           |                                                                         |
| Router# show sbc mysbc sbe adjacencies h323adj detail                 |                                                                         |

The following example lists the adjacency that is configured in the SBE using the *show sbc sbc adjacencies detail* command. The output also displays information about the H.225 messages.

**Router# show sbc mysbc sbe adjacencies h323adj detail**

SBC Service "sbc"
Adacency h323adj (H.323)
  Status: Detached
  Signaling address: 0.0.0.0:1720 (default)
  Signaling-peer: 0.0.0.0:1720 (default)
  Admin Domain: None
  Account:               
  Media passthrough: Yes
  Group:                 
  Hunting triggers: Global Triggers
  Hunting mode: Global Mode
  Technology Prefix:     
  H245 Tunnelling: Enabled
  Fast-Slow Interworking: None
  Trust-level: Untrusted
  Call-security: Insecure
  Realm: None
  Warrant Match-Order: None
  Local Jitter Ratio: 0/1000
  H225 address block: Enabled
Configuring Separate H.245 Control Channel and RAS Tech Prefix: Example

```
configure terminal
sbc mysbc
sbe
adjacency h323 h323-fxs-1b
signaling-address ipv4 88.110.128.13
signaling-port 1720
remote-address ipv4 10.0.0.0/8
signaling-peer 10.124.2.2
signaling-peer-port 1720
account h323-fxs-1b
tech-prefix 2#
h245-tunnel disable
attach
exit
```

Configuring User Protocol Timer Controls: Example

```
configure terminal
sbc mysbc
sbe
adjacency h323 abcd
adjacency timeout 10000
h225 timeout establishment 40000
adjacency timeout 10000?
h225 timeout establishment ?
  establishment h225 establishment state timeout value.
  proceeding h225 proceeding state timeout value.
  setup h225 setup timeout value.
h225 timeout proceeding 30000
h225 timeout setup 30000
ras ?
  retry RAS retry configuration.
  rrq RRQ (Registration Request) configuration.
  timeout RAS timeout configuration.
ras retry ?
  arq Retry count for an ARQ transaction.
  brq Retry count for an BRQ transaction.
  drq Retry count for an DRQ transaction.
  grq Retry count for an GRQ transaction.
  rrq Retry count for an RRQ transaction.
  urq Retry count for an URQ transaction.
ras retry arq 2
ras retry brq 2
ras retry drq 2
ras retry rrq 2
ras retry rrq 2
ras rrq ?
  keepalive Rate for keepalive msgs to refresh an H323 adjacency registration.
  ttl TTL (time to live) value for an RRQ request.
ras rrq keepalive ?
  <15000-150000> Keapalive refresh time in milliseconds - default: 45000
  ras rrq keepalive 15000
```
ras rrq ttl?
  <16-300> TTL value in seconds - default: 60
ras rrq ttl 30
adjacency timeout 30000
ras timeout?
  arq  Timeout value for an ARQ transaction.
  brq  Timeout value for an BRQ transaction.
  drq  Timeout value for an DRQ transaction.
  grq  Timeout value for an GRQ transaction.
  rrq  Timeout value for an RRQ transaction.
  urq  Timeout value for an URQ transaction.
ras timeout arq?
  <1000-45000> Timeout value in milliseconds - default: 5000
ras timeout arq 1000
ras timeout brq 1000
ras timeout drq 1000
ras timeout grq 1000
ras timeout rrq 1000
ras timeout urq 1000
Configuring User Protocol Timer Controls: Example
H.323 to SIP Interworking

The H.323 to SIP interworking capability is very important in Voice over IP (VoIP) services since both protocols are widely used in the industry. When one VoIP service provider uses Session Initiation Protocol (SIP) and another provider uses H.323, the two protocols need to interwork to enable the customers to contact each other. H.323 is an older protocol that is gradually supplanted by SIP. The customers who have their VoIP network managed using H.323 may have to transition to SIP in the future. During this transition, both protocols need to interwork on the customers' VoIP network.

The following H.323 to SIP interworking features are supported:

- H.323 to SIP Support for Emergency Calls, page 34-4
- H.323 Slow Start Calls to SIP Calls, page 34-4
- H.323 to SIP Cause Code Mapping, page 34-5
- SIP Calls to H.323 Fast Start Calls, page 34-7
- H.323 Fast Start Calls to SIP Calls, page 34-9
- SIP to H.323 Interworking for Basic Call Hold, page 34-10
- Overview: Extending the SIP Secure Calls over the H.323 Interface, page 34-11
- Configuring the SIP Secure Calls over an H.323 Interface, page 34-12
- Overview: Extending the SIP Secure Calls over the H.323 Interface, page 34-11
- Configuring the SIP Secure Calls over an H.323 Interface, page 34-12

In addition, T.38 fax passthrough is supported for SIP, H.323-H.323, and SIP-H.323 calls. See the Fax Support chapter for more information.

Note

This feature is supported in the unified model in Cisco IOS XE Release 2.5 and later.

Feature History for H.323 to SIP Interworking

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 2.5</td>
<td>H.323 to SIP interworking capability was introduced on the Cisco ASR1000 Series Aggregation Services Routers.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 3.2S</td>
<td>Allow the secure SIP calls to be interworked with the H.323 networks on the Cisco ASR1000 Series Aggregation Services Routers.</td>
</tr>
</tbody>
</table>
Contents

This module contains the following sections:

- Restrictions for H.323-SIP Interworking, page 34-2
- Information About H.323-SIP Interworking, page 34-3
- H.323 to SIP Support for Emergency Calls, page 34-4
- H.323 Slow Start Calls to SIP Calls, page 34-4
- H.323 to SIP Cause Code Mapping, page 34-5
- SIP Calls to H.323 Fast Start Calls, page 34-7
- H.323 Fast Start Calls to SIP Calls, page 34-9
- SIP to H.323 Interworking for Basic Call Hold, page 34-10
- Overview: Extending the SIP Secure Calls over the H.323 Interface, page 34-11
- Prerequisites for the SIP Secure Calls over an H.323 Interface, page 34-12
- Restrictions for the SIP Secure Calls over an H.323 Interface, page 34-12
- Configuring the SIP Secure Calls over an H.323 Interface, page 34-12
- Configuration Example: Implementing Secure SIP Calls over an H.323 Adjacency, page 34-14

Restrictions for H.323-SIP Interworking

The following features are not supported:

- Transcoding of interworking calls.
- Media bypass for interworking calls.
- H.323 DTMF signaling using any method other than the alphanumeric method of UserInputIndication.
- Interworking of endpoint registrations (not supported by H.323).
- Failover of interworking calls (because H.323 call legs cannot be preserved across a failover).
- Interworking of any SIP method other than INVITE, ACK, CANCEL, BYE, INFO, or PRACK.
- End-to-end authentication on an interworking call. For example, an H.323 call branch cannot challenge a SIP call branch and vice versa. The Session Border Controller (SBC) itself can challenge a SIP call branch, but not an H.323 call branch.
- User-configurable mapping of cause codes.
- User-configurable mapping of codec types.
- Interworking of signaling support for Silence Suppression/VAD. It is assumed that the majority of endpoints interoperate correctly without explicitly signaling silence suppression.
- Interworking of video calls.
- Interworking of fax calls partially. T.38 fax is supported for SIP, H.323-H.323, and SIP-H.323 calls.
- Payload interworking is not supported.
Information About H.323-SIP Interworking

Following the usual process, after the SBC applies the call and number policy tables, a final adjacency and account are chosen. In H.323 to SIP interworking, the originating and terminating adjacencies are configured for different protocols. For example, the originating adjacency can be configured for H.323 and the terminating adjacency can be configured for SIP.

H.323 has two modes of operation: slow start and fast start. The initiation of a call may proceed in a slow start or fast start in H.323. In a slow start, H.323 signaling consists of Setup, Call Proceeding, Alerting, and Connect steps. After these steps, the H.245 media negotiation is performed.

When a call is initiated in H.323 fast start, the H.245 media negotiation is performed within the initial Setup message.

The SBC supports the following features of H.323 to SIP and SIP to H.323 interworking:

- **SIP upstream, H.323 fast-start downstream**, offer received on the SIP INVITE. See *Cisco Unified Border Element (SP Edition) supports interworking of upstream SIP endpoints calling to downstream H.323 Fast Start endpoints. This support includes support for Early Media*, page 34-7.

- **SIP upstream, H.323 slow-start downstream**, offer received on the SIP INVITE. First, H.323 fast-start is tried downstream. The SBC drops back to slow-start procedures when it discovers the downstream endpoint does not support fast-start. See *H.323 Slow Start Calls to SIP Calls*, page 34-4.

- **SIP upstream, H.323 downstream (either fast-start or slow-start)**, no offer received on the SIP INVITE. See *SIP Calls to H.323 Fast Start Calls*, page 34-7.

- **H.323 fast-start upstream, SIP downstream**. See *H.323 Fast Start Calls to SIP Calls*, page 34-9.

- **H.323 slow-start upstream, SIP downstream**. SIP downstream is tried with a default SDP offer, containing a single media channel with the following offered codecs in decreasing order of preference: G.729, G.711 U-law, G.711 A-law, and G.723. See the *H.323 Slow Start Calls to SIP Calls? section on page 34-4*.

- **Mapping of SIP response codes to H.225 error codes used by H.323 and mapping of H.225 error codes to SIP response codes**. See *H.323 to SIP Cause Code Mapping*, page 34-5.

- **Interworking for basic call hold feature to translate, hold, and resume signaling in H.323 and SIP interworking calls**. See *SIP to H.323 Interworking for Basic Call Hold*, page 34-10.

- **Early media in SIP calls to H.323 fast start calls**. See *Early Media Support*, page 34-8.

- **DTMF interworking between SIP and H.323 in the signaling plane, using the alphanumeric method of UserInputIndication**.

\[\text{Note}\]

All H.323 calls, including established H.323-H.323 and SIP-H.323 interworking calls, are disconnected upon an SBC switchover. An SBC switchover occurs when an active RP switches over to the standby RP in a hardware redundant system (such as a Cisco ASR 1006 Router) or when the active IOS process switches over to the standby IOS process in a redundant software system (such as a Cisco ASR 1002 Router).

When you are configuring the SBC to interwork calls between H.323 and SIP networks, you can also consider the following configuration tasks:

- **For networks that use RFC2833 telephone-event signaling**, you may want to configure telephone-event support on the H.323 or SIP side for improved call setup efficiency.
For DTMF interworking with H.323-SIP calls, you may want to configure the telephone-event payload type supported by the caller and callee through Call Admission Control (CAC) policy. This allows for improved call setup efficiency.

To allow pass-through of display name updates, for example, following a third-party call transfer—you may want to whitelist the SIP Remote-Party-ID header.

**H.323 to SIP Support for Emergency Calls**

Cisco Unified Border Element (SP Edition) supports H.323 to SIP call routing for emergency calls. Cisco Unified Border Element (SP Edition) routes voice and video calls according to the configured session routing policy. A call is categorized as “emergency” based on the dialed number or on the Resource-Priority header if it is originated on the SIP side. Based on the emergency categorization, special routing and Call Admission Control (CAC) logic is applied.

**H.323 Slow Start Calls to SIP Calls**

Cisco Unified Border Element (SP Edition) supports interworking of H.323 Slow Start upstream calls made to a SIP endpoint.

As a result of a H.323 Slow start call, the downstream SIP endpoint has no media information from the calling endpoint at the time it sends the initial SIP INVITE. The SBC has the ability to send a default session description protocol (SDP) offer on the INVITE that proposes a single media stream for voice traffic, listing the following candidate codecs in decreasing order of preference: G.729, G.711 U-law, G.711 A-law, and G.723.

The answer received on the 200 OK response may have either reduced the number of codecs for the stream to 1 (called a “mono-answer”), or the 200 OK response still has multiple codecs in the stream (called a “multi-answer”). The “multi-answer” is unacceptable to the H.323 protocol. The SBC has the ability to refine the codec list by making a re-INVITE to the SIP endpoint containing only the first codec. Figure 34-1 shows the flows that take place during this process.
H.323 to SIP Cause Code Mapping

Cisco Unified Border Element (SP Edition) supports mapping of SIP response codes to H.225 error codes used by H.323 and mapping of H.225 error codes to SIP response codes.

In H.323 to SIP interworking, the SBC provides call rejection with the proper cause code in the following manner:

- If a downstream SIP endpoint rejects a call, the response is translated into the H.225 error code set for the upstream H.323 device. The SIP endpoint may also reject attempts by the SBC to refine the codec list.
- If a downstream H.323 endpoint rejects a call, there are two possible actions—the H.323 gatekeeper may reject admission for the call, or the endpoint sends a Release Complete to reject the call.
Table 34-1 shows how SIP response codes are mapped to H.225 error codes.

<table>
<thead>
<tr>
<th>SIP error code in</th>
<th>H.225 error code out</th>
</tr>
</thead>
<tbody>
<tr>
<td>301</td>
<td>UnreachableDestination</td>
</tr>
<tr>
<td>302</td>
<td>UnreachableDestination</td>
</tr>
<tr>
<td>400</td>
<td>Undefined Reason</td>
</tr>
<tr>
<td>401</td>
<td>No Permission</td>
</tr>
<tr>
<td>403</td>
<td>Security Denied</td>
</tr>
<tr>
<td>404</td>
<td>UnreachableDestination</td>
</tr>
<tr>
<td>405</td>
<td>Undefined Reason</td>
</tr>
<tr>
<td>406</td>
<td>Undefined Reason</td>
</tr>
<tr>
<td>407</td>
<td>No Permission</td>
</tr>
<tr>
<td>408</td>
<td>Undefined Reason</td>
</tr>
<tr>
<td>410</td>
<td>Unreachable Destination</td>
</tr>
<tr>
<td>413</td>
<td>Undefined Reason</td>
</tr>
<tr>
<td>414</td>
<td>Undefined Reason</td>
</tr>
<tr>
<td>415</td>
<td>Undefined Reason</td>
</tr>
<tr>
<td>416</td>
<td>Undefined Reason</td>
</tr>
<tr>
<td>420</td>
<td>Undefined Reason</td>
</tr>
<tr>
<td>421</td>
<td>Undefined Reason</td>
</tr>
<tr>
<td>423</td>
<td>Undefined Reason</td>
</tr>
<tr>
<td>480</td>
<td>Destination Rejection</td>
</tr>
<tr>
<td>481</td>
<td>Unreachable Destination</td>
</tr>
<tr>
<td>482</td>
<td>Undefined Reason</td>
</tr>
<tr>
<td>483</td>
<td>Undefined Reason</td>
</tr>
<tr>
<td>487</td>
<td>Destination Rejection</td>
</tr>
<tr>
<td>488</td>
<td>Undefined Reason</td>
</tr>
<tr>
<td>501</td>
<td>Undefined Reason</td>
</tr>
<tr>
<td>503</td>
<td>Undefined Reason</td>
</tr>
<tr>
<td>504</td>
<td>Undefined Reason</td>
</tr>
<tr>
<td>505</td>
<td>Undefined Reason</td>
</tr>
<tr>
<td>513</td>
<td>Undefined Reason</td>
</tr>
<tr>
<td>603</td>
<td>Undefined Reason</td>
</tr>
<tr>
<td>604</td>
<td>Unreachable Destination</td>
</tr>
<tr>
<td>606</td>
<td>Undefined Reason</td>
</tr>
</tbody>
</table>
Table 34-2 shows how H.225 error codes are mapped to SIP response codes.

<table>
<thead>
<tr>
<th>H.225 error code in</th>
<th>SIP error code out</th>
</tr>
</thead>
<tbody>
<tr>
<td>NoBandwidth</td>
<td>500</td>
</tr>
<tr>
<td>UnreachableDestination</td>
<td>604</td>
</tr>
<tr>
<td>DestinationRejection</td>
<td>486</td>
</tr>
<tr>
<td>No Permission</td>
<td>401</td>
</tr>
<tr>
<td>GatewayResource</td>
<td>503</td>
</tr>
<tr>
<td>BadFormatAddress</td>
<td>404</td>
</tr>
<tr>
<td>SecurityDenied</td>
<td>403</td>
</tr>
<tr>
<td>InvalidRevision</td>
<td>503</td>
</tr>
<tr>
<td>UnreachableGatekeeper</td>
<td>503</td>
</tr>
<tr>
<td>AdaptiveBusy</td>
<td>503</td>
</tr>
<tr>
<td>InConf</td>
<td>503</td>
</tr>
<tr>
<td>RouteCallToGatekeeper</td>
<td>503</td>
</tr>
<tr>
<td>CallForwarded</td>
<td>503</td>
</tr>
<tr>
<td>RouteCallToMC</td>
<td>503</td>
</tr>
<tr>
<td>FacilityCallDeflection</td>
<td>503</td>
</tr>
<tr>
<td>CalledPartyNotRegistered</td>
<td>503</td>
</tr>
<tr>
<td>CallerNotregistered</td>
<td>503</td>
</tr>
<tr>
<td>ConferenceListChoice</td>
<td>503</td>
</tr>
<tr>
<td>StartH245</td>
<td>503</td>
</tr>
<tr>
<td>NewConnectionNeeded</td>
<td>503</td>
</tr>
<tr>
<td>NoH245</td>
<td>503</td>
</tr>
<tr>
<td>NewTokens</td>
<td>503</td>
</tr>
<tr>
<td>FeatureSetUpdate</td>
<td>503</td>
</tr>
<tr>
<td>ForwardedElements</td>
<td>503</td>
</tr>
<tr>
<td>TransportedInformation</td>
<td>503</td>
</tr>
</tbody>
</table>

**SIP Calls to H.323 Fast Start Calls**

Cisco Unified Border Element (SP Edition) supports interworking of upstream SIP endpoints calling to downstream H.323 Fast Start endpoints. This support includes support for Early Media.
Early Media Support

The capability for SIP endpoints calling to H.323 Fast Start endpoints includes support for Early Media, where the Fast Start response may be received before the Connect. Early Media can flow when the caller (SIP endpoint) makes a media proposal on the initial call setup request and the callee (the H.323 endpoint) responds to the offer before the call is connected. In this case, the H.323 endpoint expects an SDP offer on the initial INVITE.

H.323 may send a “Progress Indicator” on any H.225 message that it sends the SBC. A progress indicator of a value of 1 or 8 indicates that the H.323 endpoint will send early media. In an interworking call, only the first Progress Indicator received from the H.323 endpoint is acted upon.

If the H.323 endpoint sends a progress indicator with a value of 1 or 8, then in an interworking call with a SIP upstream call, if sufficient media parameters have been negotiated with the H.323 endpoint, the SBC returns a 183 provisional response to the SIP caller with the SDP indicating early media.

Depending on the Call Admission Control (CAC) configuration, the SBC may allow early media to be passed through at this point. If insufficient media parameters have been received to build the SDP to send to the SIP endpoint, then the SBC waits for media negotiation with the H.323 endpoint to reach a point where the SDP can be generated and then the SBC can send the 183 provisional response. Figure 34-2 shows the flows that take place during this process.

![Figure 34-2 SIP with SDP Offer Call to H.323 Fast Start](image-url)
Cisco Unified Border Element (SP Edition) supports interworking of H.323 Fast Start upstream calls made to a downstream SIP endpoint.

If the Fast Start offer from the H.323 device includes alternative codec options, the SDP offer sent to the downstream SIP device lists all of these alternative codecs in the same order of preference as that supplied by H.323. The most preferred codec is listed first. If the SIP endpoint accepts more than one codec, this is not acceptable on the H.323 Fast Start Response. Therefore, the SBC is able to refine the offer. The SBC makes another offer to the SIP device with a single codec option, by taking the most preferred codec listed in the SDP answer and constructing a new offer with that codec in it. If the downstream SIP device accepts this offer, then a FastStart response is returned, selecting that codec.

If the SIP endpoint SDP does not need to be refined, the fast start response will go back on the first available message to the H.323 endpoint.

The SIP endpoint may send early media as soon as it has sent the multi-SDP answer. However, the early media will fail to get through until the mono-SDP answer is received and processed, and the Progress Indicator and Fast Start response sent to the H.323 endpoint. See the Early Media Support section on page 34-8.

Figure 34-3 shows the flows that take place during this process.
SIP to H.323 Interworking for Basic Call Hold

The SIP/H.323 interworking for basic call hold feature enables Cisco Unified Border Element (SP Edition) to translate, hold, and resume signaling in H.323 and SIP interworking calls.

Note
Basic call hold does not require external configuration and is enabled by default.

SIP Requirements

In RFC-3264 SDP Offer-Answer protocol, basic call hold is signaled by a re-Offer that includes an a=sendonly', 'a=inactive', or 'c=IN IP4 0.0.0.0' line.

- a=sendonly or c=IN IP4 0.0.0.0 indicates that the offerer wants to keep transmitting. The Answer may optionally force the offerer to cease transmitting by setting a=inactive or c=IN IP4 0.0.0.0.
- a=inactive indicates that the offerer will also cease transmitting. In this case, the answerer must also reply with a=inactive.

Resume is signaled by setting the direction to a=sendrecv or, because this is the default setting, omitting the direction line altogether.

For SIP, requirements are:

- The SBC must support receipt of all of the above forms of call hold signaling. On transmit, control should preferably be provided over the form that is used.
- Translation of a re-offer that opens or closes the send direction (not just the receive direction).
- Case of the offerer or answerer changing their RTP address/port on a call hold resume offer or a call hold answer.
- Sending a re-Offer on a SIP re-INVITE and processing the answer on the INVITE 200 rsp.
- Processing an incoming answer on the first re-INVITE response even if that is not the final response (In this case, a duplicate answer on the final 200 response must be ignored).
- Receipt of a re-offer on a SIP INVITE request.
- Sending an answer on a re-INVITE 200 response.

H.323 Requirements

In H.245, basic call hold is signaled by sending an empty terminal capability set (defined in H.323 section 8.4.6, and known as "TCS=0" or "ECS"). The receiver of the TCS=0 must close its send channel and avoid re-opening it. Resume is signaled by sending a non-empty terminal capability set. At this point, the send channel is re-opened. In terms of the H.245 message flows:

- Terminal capabilities are transmitted using a TerminalCapabilitySet (TCS). This message is responded to with a TerminalCapabilitySetAck (TCS Ack) or TerminalCapabilitySetReject.
- A channel is opened with an H.245 OpenLogicalChannel (OLC). This is responded to with an OpenLogicalChannelAck (OLC Ack) or OpenLogicalChannelReject.
- A channel is closed with an H.245 CloseLogicalChannel (CLC). A CloseLogicalChannelAck (CLC Ack) indicates that this message has been processed.

For H.323, requirements are:
Overview: Extending the SIP Secure Calls over the H.323 Interface

Data security has become the prime objective enhances service providers, corporates, and government institutes. Cisco IOS XE Release 3.2S enhances the security feature, by extending support to the secure calls coming from either a H323 adjacency or a SIP adjacency. Before this enhancement, the SBC supported only the SIP secure calls, and the SIP secure calls were not able to interwork with the H.323 networks. After this enhancement, the SIP secure calls received from a SIP adjacency and routed over an H323 adjacency can be sent by configuring the corresponding H323 adjacency as trusted. Also, calls coming from H323 adjacency can be configured as secure calls.
To configure an H.323 adjacency as trusted for handling the SIP secure calls received from a SIP adjacency, use the `trunk trusted` command. Defining an adjacency as trusted, distinguishes it from untrusted adjacencies. If an incoming call is a secure call, it goes through trusted adjacency. If no trusted adjacencies are configured, the incoming secure call is rejected with the SIP response code 403 (Forbidden) or an H.225 with the reason as Security Denied. If the incoming call is not a secure call, it can go through a trusted adjacency or untrusted adjacency.

To handle the calls coming from H.323 adjacency and to treat them as secure calls, configure the H.323 adjacency as secure using the `inbound secure` command. The outgoing SIP calls become a SIP-secure calls.

Prerequisites for the SIP Secure Calls over an H.323 Interface

Following are the prerequisites for the SIP Secure calls over an H.323 interface:

- The minimum software image required for this feature to work is the Cisco IOS XE 3.2S Software image.
- An H.323 adjacency must be configured as trusted before configuring the incoming calls as secure.

Note: All the H.323 adjacencies that are defined are by default untrusted. If you want to change an adjacency from trusted to untrusted, configure the incoming calls for the adjacency as insecure by using the `no inbound secure` command.

Restrictions for the SIP Secure Calls over an H.323 Interface

Following are the restrictions of the SIP Secure calls over an H.323 interface:

- The SBC does not signal secure H.323 calls using the procedure described in H.235. It also does not recognize the secure nature of the incoming H.323 calls using the H.235 procedures.
- The SBC does not use a TLS or IPSec to send call signalling for secure H.323 calls.

Configuring the SIP Secure Calls over an H.323 Interface

To implement the SIP Secure calls over an H.323 interface, configure the following:

- An H.323 outgoing adjacency as Trusted for handling the SIP secure calls received from the SIP adjacency.
- Incoming calls from an H.323 adjacency as secure calls for calls coming from an H.323 adjacency.

The following section provides procedure for configuring an H.323 adjacency as trusted and configuring incoming calls from an H.323 adjacency as secure calls:

SUMMARY STEPS

1. configure terminal
2. sbc sbcname
3. sbe
### Chapter 34      H.323 to SIP Interworking

#### Configuring the SIP Secure Calls over an H.323 Interface

4. adjacency h323 adjacency-name
5. trunk trusted
6. inbound secure

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure terminal</td>
<td>Enters the global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>sbc service-name</td>
<td>Enters the mode of an SBC service.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>Use the service-name argument to define the name of the service.</td>
</tr>
<tr>
<td></td>
<td>Router(config)# sbc mysbc</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>sbe</td>
<td>Enters the mode of an SBE entity within an SBC service.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>adjacency h323 adjacency-name</td>
<td>Enters the H.323 adjacency mode to configure the parameters for the specified adjacency name.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe)# adjacency h323 trust-h323-adj</td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td>trunk trusted</td>
<td>Configures the H.323 adjacency as trusted.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-adj-h323)# trunk trusted</td>
<td></td>
</tr>
<tr>
<td>Step 6</td>
<td>inbound secure</td>
<td>Configures the incoming calls from the H.323 adjacency as secure calls.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-adj-h323)# inbound secure</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Note</td>
<td>If the H.323 adjacency is configured as untrusted, incoming calls cannot be configured as secure calls.</td>
</tr>
</tbody>
</table>
Configuration Example: Implementing Secure SIP Calls over an H.323 Adjacency

The following example shows how to configure an H.323 adjacency as Trusted and mark the incoming calls on an H.323 adjacency as secure calls:

Router# configure terminal
Router(config)# sbc mysbc
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency h323 trust-h323-adj
Router(config-sbc-sbe-adj-h323)# trunk trusted
Router(config-sbc-sbe-adj-h323)# inbound secure

The following example displays the configuration details of trust-h323-adj:

Router# show sbc mysbc sbe adjacencies trust-h323-adj detail

SBC Service "mysbc"
Adjacency trust-h323-adj (H.323)
Status: Detached
Signaling address: 0.0.0.0:1720 (default)
Signaling-peer: 0.0.0.0:1720 (default)
Admin Domain: None
Account:
Media passthrough: Yes
Group:
Hunting triggers: Global Triggers
Hunting mode: Global Mode
Technology Prefix:
H245 Tunnelling: Enabled
Fast-Slow Interworking: None
Trust-level: Trusted
Call-security: Secure
Realm: None
Warrant Match-Order: None
Support for H.239

H.239 is an extension to the H.323 family of specifications to allow a second video stream in parallel with the primary live video stream to share any type of content such as slides and spreadsheets. This second stream is one-way and considered important in video-conferencing where a viewer can see the speaker and in parallel, the presentation slides. This mode of the conference is controlled by an Multipoint Control Unit (MCU).

Cisco Unified Border Element (SP Edition) was earlier known as Integrated Session Border Controller, and is referred to as SBC in this document.


For information about all the Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the Cisco IOS master commands list.

Feature History of Support for H.239 on the Cisco Unified Border Element (SP Edition)

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 3.3S</td>
<td>The Support for H.239 feature was introduced on the Cisco ASR 1000 Series Routers.</td>
</tr>
</tbody>
</table>

Contents

- Information on Support for H.239, page 35-1
- Restriction for Support for H.239, page 35-2

Information on Support for H.239

H.245-based systems provide for multiple channels of video, while H.320 systems provide for only a single video channel. However, neither of these define a one-way transmission method, methods to label a video channel’s content as a presentation video stream, or methods to control presentation video in a multipoint conference. H.239 provides these extensions, along with the ability to add an additional video channel to H.320.
It defines new capabilities, such as H239ControlCapability and H239ExtendedVideoCapability, which a terminal can advertise in its Terminal Capability Set (TCS) message. These capabilities indicate that the terminal can support H.239, and the roles, presentation with slides or live video, that the terminal can support. The SBC passes these capabilities between the H.323 endpoints without requiring to understand them.

The Support for H.239 feature defines a number of H.239 messages that can be carried within H.245 generic messages. These messages are used to negotiate tokens and general flow control. The MCU provides the token on request by a terminal to send a slide-set. This process ensures that at any point only one terminal within the conference is sending a slide-set because a terminal has to wait until the MCU provides the token before initiating the new video channel.

Once a terminal has a token, it can open an additional video channel to carry the presentation stream. This can be done using the standard H.245 messages such as OpenLogicalChannel. However, the OpenLogicalChannel message for the presentation stream has an additional h239ExtendedVideoCapability block, indicating the role of this video stream. The SBC is enhanced to understand the h239ExtendedVideoCapability block.

**Note**
An endpoint can choose to originate multiple additional video channels.

### Restriction for Support for H.239

The Support for H.239 feature has the following restrictions:

- Interworking SIP-H.323 Video calls using H.239 is not supported.
- Redundancy for H.323 calls is not supported.
- A fast-start request cannot include a request to open an H.239 additional video channel as it is not supported.
- H.239 systems based on H.235 is not supported.
- The SBC does not support call transfer for H.323 calls. When an H.323 endpoint is placed on hold, it closes its media as well as video channels.
Implementing Billing on Cisco Unified Border Element (SP Edition)

The Cisco Unified Border Element (SP Edition) billing component includes the following core features:

- Compatibility with existing billing systems—To be able to fit the Cisco Unified Border Element (SP Edition) billing system easily into an existing billing architecture of a provider is an important functional requirement. This requirement entails the use of mechanisms to obtain billing information in a similar fashion to those of the existing mechanisms.

- Integration with next-generation technologies and solutions—Equally important is the requirement to use next-generation billing technologies, so that service information from Cisco Unified Border Element (SP Edition), softswitches, voicemail, and unified messaging applications, and so on can be collated and billed in a distributed environment.

- High availability and fault tolerance.

- Flexible architecture.

The billing component functions as a third-party integrated, distributed Remote Authentication Dial-In User Service (RADIUS)-based call and event logging.

The function of the billing component is:

- Third-party integrated, distributed Remote Authentication Dial-In User Service (RADIUS)-based call and event logging.

Note
This feature is supported in the unified model for Cisco IOS XR Software Release 2.4 and later.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).


For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.
Feature History for Implementing Cisco Unified Border Element (SP Edition) Billing

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 2.4</td>
<td>This feature was introduced on the Cisco IOS XR along with support for the unified model.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 2.5</td>
<td>Support for Media Information and Granular Timestamp Support were added on the Cisco ASR 1000 Series Router.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 2.6.1</td>
<td>Support for Adjacency Information was added on the Cisco ASR 1000 Series Router.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 2.6.2</td>
<td>Support for Endpoint Information was added on the Cisco ASR 1000 Series Router.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 3.2.0S</td>
<td>Support for XML based billing method was introduced on the Cisco ASR 1000 Series Router.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 3.3S</td>
<td>The Selective RADIUS Billing feature was added on the Cisco ASR 1000 Series Router.</td>
</tr>
</tbody>
</table>

Contents

This module contains the following sections:

- Prerequisites for Implementing Billing, page 36-2
- Information About Implementing Billing, page 36-3
- Support for Local Cache, page 36-6
- Support for Media Information, page 36-6
- How to Implement Billing, page 36-7
- Configuration Examples of Implementing Billing, page 36-13
- Selective RADIUS Billing, page 36-20
- Configuration Example of Selective RADIUS Billing, page 36-22

Prerequisites for Implementing Billing

The following prerequisites are required to implement Cisco Unified Border Element (SP Edition) billing:

- Before implementing interworking billing, Cisco Unified Border Element (SP Edition) must already be configured.
- To implement billing on the signaling border element (SBE) you must obtain a unique network element ID for the SBE from your network administrator. In addition, you must perform the following task depending on what form of billing you require.
  - To implement integrated RADIUS-based call logging, you must first configure the RADIUS server and set up the RADIUS network infrastructure.
Information About Implementing Billing

It is critical to understand all Cisco Unified Border Element (SP Edition) billing features and capabilities before performing billing configurations for the Cisco Unified Border Element (SP Edition). The following sections describe Cisco Unified Border Element (SP Edition) billing topologies:

- Integrated Billing Systems, page 36-3
- Granular Timestamp Support, page 36-4

Integrated Billing Systems

Integrated billing is achieved through the PacketCable Event Messages architecture (Figure 36-1 shows the PacketCable 1.5 Event Messages Specification; PKT-SP-EM1.5-I01-050128) where the Cisco Unified Border Element (SP Edition) is integrated into this architecture. As shown, the billing server and softswitch both support PacketCable Event Messages.

ISP-A in Figure 36-1 shows Cisco Unified Border Element (SP Edition) operating in a unified model where the billing system is being deployed as a distributed billing system consisting of three billing servers. Cisco Unified Border Element (SP Edition) can be configured to send to these servers in a range of ways, such as to all three simultaneously, or to use one primary and two backups.

In the unified model, the system operates as follows:

- Cisco Unified Border Element (SP Edition) produces event messages (EMs). These event messages are for billable or other interesting events, such as call start, call end, and media-type changes.
- Cisco Unified Border Element (SP Edition) and other elements of the system, which produces EMs, sends them in real time (or batched up for network efficiency) using the RADIUS protocol to the billing server.
- Billing server collates EMs into call detail records (CDRs). For an example of a CDR, see Example for Event Messages from Cisco Unified Border Element (SP Edition) to RADIUS Billing Server? section on page 37-14.
- Cisco Unified Border Element (SP Edition) supports local caching of records and event messages in the Cisco ASR 1000 Series Router’s local disk in the event that none of the RADIUS servers are reachable.
- Cisco Unified Border Element (SP Edition) supports multiple RADIUS servers, for example, you can define multiple servers under a single client.

Note that ISP-B in Figure 36-1 shows Cisco Unified Border Element (SP Edition) operating in a distributed model where the billing system is being deployed using a single billing server and a softswitch.

Table 36-1 shows the packet billing termination codes that are supported by Cisco Unified Border Element (SP Edition).

<table>
<thead>
<tr>
<th>Code Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0003</td>
<td>No route to destination</td>
</tr>
<tr>
<td>0016</td>
<td>Normal call clearing</td>
</tr>
<tr>
<td>0017</td>
<td>User busy</td>
</tr>
<tr>
<td>0019</td>
<td>User alerting: No answer</td>
</tr>
</tbody>
</table>
Table 36-1  Supported Packet Billing Termination Codes (continued)

<table>
<thead>
<tr>
<th>Code Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0020</td>
<td>Subscriber absent</td>
</tr>
<tr>
<td>0027</td>
<td>Destination out of order</td>
</tr>
<tr>
<td>0028</td>
<td>Invalid number format (incomplete address)</td>
</tr>
<tr>
<td>0031</td>
<td>Unknown: Call ended during recovery processing</td>
</tr>
<tr>
<td>0041</td>
<td>Temporary failure</td>
</tr>
<tr>
<td>0042</td>
<td>Switching equipment congestion</td>
</tr>
<tr>
<td>0047</td>
<td>Resource unavailable, unspecified</td>
</tr>
<tr>
<td>0063</td>
<td>Service or option not available, unspecified</td>
</tr>
<tr>
<td>0065</td>
<td>Bearer capability not implemented</td>
</tr>
<tr>
<td>0095</td>
<td>Invalid message, unspecified</td>
</tr>
<tr>
<td>0097</td>
<td>Message type nonexistent or not implemented</td>
</tr>
<tr>
<td>0099</td>
<td>Information element nonexistent or not implemented</td>
</tr>
<tr>
<td>0103</td>
<td>Parameter non-existent or not implemented, passed on</td>
</tr>
<tr>
<td>0111</td>
<td>Protocol error: Unspecified</td>
</tr>
<tr>
<td>0127</td>
<td>Interworking: Unspecified</td>
</tr>
</tbody>
</table>

Note

The PacketCable 1.5 Event Messages Specification discusses sending the identifying information (the BCID and FEID) on the outgoing INVITE and responding SDP so that correlation can be done between the two sets of billing data. Cisco Unified Border Element (SP Edition) does not support this mechanism for intra-domain or inter-domain transmission. The billing server must perform the correlation using an alternative method (for example, using the telephone numbers dialed and the time of the call).

Granular Timestamp Support

Cisco Unified Border Element (SP Edition) Billing Manager maintains a granular timestamp that billing methods can use to query the current time. The granular timestamp provides a precision of 100 milliseconds. This precision is sufficient for all billing requirements without having an impact on performance.

By default, the granular timestamp is set to the maximum of 100 milliseconds.

Endpoint Information in PacketCable Billing

Beginning Cisco Unified Border Element (SP Edition) Release 2.6.2, you can configure SBC to include information—adjacency name or addressing—on endpoints in use for a given call in the PacketCable billing records.
When SBC is not configured to include the endpoint information in the messages, the Signaling_Start messages for both sides of a call contains an MTA_Endpoint_Name attribute that contains the string MTA Endpoint. MTA_Endpoint_Name attribute is not included in Call Answer or Signaling_Stop Event Messages.

If you configure SBC to include the adjacency name, only the names of the endpoint adjacencies are included in the billing records. For example, SIPPB. If you have configured SBC to include the endpoint addressing information, then IP address, port, and transport type are also included in the billing records along with the adjacency name in the following format: IP address,port,transport type,adjacency name. For example, 2.0.0.36,5078,UDP,SIPPB.

If SBC is configured to include the endpoint information:

- SBC adds the source adjacency name or addressing information, as configured, to the upstream Signaling_Start Event Messages. This information is included in the MTA_Endpoint_Name attribute—replacing the hard-coded string MTA Endpoint. At this point, the downstream Signaling_Start message contains only the hard-coded string—MTA Endpoint.
- SBC adds the destination adjacency name or addressing information, as configured, to downstream Call_Answer Event Messages. This information is included in the MTA_Endpoint_Name attribute. The upstream Call_Answer message does not contain an MTA_Endpoint_Name attribute.
- SBC includes both—source and destination—endpoint details in the Signaling_Stop messages; the source adjacency name or addressing information in the upstream message, and the destination adjacency name or addressing information in the downstream message. This ensures that even if a call fails to connect, the billing server still has access to both endpoint details.

Use the [no] cdr endpoint-info {addressing | adjacency} command to configure SBC to include the endpoint information in PacketCable billing.

Use the show sbc sbcname sbe billing instance command to verify whether the SBC is configured to include the endpoint information.

**Restrictions for PacketCable Billing**

H.323 is supported for PacketCable billing, but with some limitations. One such limitation is that no H.323 signalling address is present in PacketCable billing.

**Performing ISSU for Endpoint Information**

When performing ISSU to upgrade SBC from Release 2.6.1 to Release 2.6.2, if adjacency information (cdr adj-info) is provisioned for Release 2.6.1, then the corresponding endpoint-info adjacency (cdr endpoint-info adjacency) option is provisioned and the functionality is maintained.

When performing ISSU to downgrade SBC from Release 2.6.2 to Release 2.6.1, if endpoint adjacency information (cdr endpoint-info adjacency) is provisioned for Release 2.6.2, then the corresponding adjacency (cdr adj-info) option is provisioned and the functionality is maintained. If endpoint addressing information (cdr endpoint-info addressing) is provisioned for Release 2.6.2, no provisioning happens in Release 2.6.1.
Support for Local Cache

The Cisco ASR 1000 Series Routers have a local disk where records and event messages (EMs) can be stored on a local cache. Local cache support is a significant advantage because call detail records and EMs are not lost when a billing server is unavailable. Use the `cache` command to configure parameters for storing call detail records and EMs on local disk.

In a typical integrated billing environment, as calls come up and go down, billing records are generated and sent to the RADIUS server. When for any reason the RADIUS server is not reachable or not responding to accounting packets, then the Billing Manager marks the transport as DOWN. As soon as the transport goes down and the local caching path is defined with the `cache path` command, the billing records are cached locally on the Cisco ASR 1000 Series Router disk. Your router disk may be the hard disk, bootflash or usb0, depending on router configuration. Subsequently, every 10 seconds, the Cisco ASR 1000 Series Router tries to send the cached information to the RADIUS server.

Support for Media Information

Cisco Unified Border Element (SP Edition) supports reporting media information in billing messages. The PacketCable event message (EM) billing interface reports the properties of the media streams associated with a call, including when the media stream begins and ends, the packets and octets transmitted, and lost latency and jitter statistics.

The Support for Media Information feature defines a new proprietary RADIUS Vendor-Specific Attribute that can be carried on the QoS_Commit and QoS_Release PacketCable messages. This attribute added to these billing messages makes stream creation information available to PacketCable billing.

Use the `cdr media-info` command to add the RADIUS Vendor-Specific Attribute to the billing messages. The RADIUS Vendor-Specific Attribute contains the following information:

- Local IP address and port and remote media endpoint IP address and port used in the media stream.
- Direction of the media stream (send-only, receive-only, send-and-receive, or inactive).
- Codecs negotiated for that media stream.
- Bandwidth reserved for the media stream.

Restrictions for Support for Media Information

The restriction for Support for Media Information are the following:

If an endpoint is behind a NAT, then the endpoint IP address cannot be obtained from the Session Description Protocol (SDP). It is instead auto detected after the endpoint sends media packets. This means that the remote address and port may not be known at the point that the gate is committed. Therefore, this information is not available on the Media_Session_Desc attribute that is sent on the QoS_Commit PacketCable message. Instead, a zero address is specified.

In particular, in a normal call setup and teardown when an endpoint is behind a NAT, there is no remote address or port in the Media_Session_Desc sent on the QoS_Commit message. The correct remote address and port is in the Media_Session_Desc sent in the QoS_Release message.

The only case in which Cisco Unified Border Element (SP Edition) would never report a remote address and port is when the call ends before any media packets have been sent and therefore the remote address is never learned by the media forwarding component on the network processing unit.
Figure 36-1 shows the PacketCable 1.5 Event Messages Specification (PKT-SP-EM1.5-I01-050128).

How to Implement Billing

The SBE can perform billing. The key objects to be configured for billing are the long duration checks and the physical location of the cache. You can configure up to eight PacketCable-EM billing instances (indexed 0-7).

Follow the procedure in the Chapter 37 Configuring Billing section on page 36-8.
Restrictions for Billing

The restrictions for configuring billing are:

- You may not modify any billing configuration items if billing is active.
- You may only modify batch-time and batch-size when a method or the billing is active. All other commands are not allowed. However, those are blocked when more than one method exists.
- You may not modify ldr-check at billing level if any methods have been defined.
- You may not remove a RADIUS accounting client if it is currently assigned to a billing method.
- You must define a RADIUS accounting client before it is selected in a billing method.
- You can assign a RADIUS accounting client only to a single billing method.
- You cannot remove the billing when it is active or when methods are configured.
- You may not remove the method packetcable command while a packetcable-em configuration is in place.
- H.323 is supported for billing, but with some limitations. One such limitation is that no H.323 signalling address is present in billing instances.

Configuring Billing

This task defines how to configure billing configurations.

SUMMARY STEPS

1. configure
2. sbc service-name
3. sbe
4. control address aaa ipv4 IP_address
5. radius accounting client-name
6. concurrent-requests num
7. retry-interval num
8. retry-limit num
9. server server-name
10. address ipv4 A.B.C.D.
11. priority pri
12. key key
13. port port-num
14. exit
15. activate
16. exit
17. billing
18. cdr endpoint-info addressing
19. ldr-check \( \{ HH \text{ MM} \} \)
20. local-address ipv4 \( \{ A.B.C.D. \} \)
21. method packetcable-em
22. cache \( \{ \text{path} \ \{ \text{WORD} \} \} \) \( | \) alarm \( \{ \text{critical VAL} \} \) \( | \) major VAL \( | \) minor VAL \( | \) max-size \( \{ 0-4194303 \} \) \}
23. packetcable-em method-index transport radius RADIUS-client-name
24. batch-size number
25. batch-time number
26. attach
27. exit
28. activate

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc service-name</td>
<td>Enters the mode of an SBC service. Use the service-name argument to define the name of the service.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# sbc mySBC</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Enters the mode of an SBE entity within an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> control address aaa ipv4 IP_address</td>
<td>Configures an SBE to use a given IPv4 AAA control address when contacting an authentication or billing server. This address is a unique address within the signaling address.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe)# control address aaa ipv4 192.168.113.2</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> radius accounting client-name</td>
<td>Enters the mode for configuring a RADIUS client for accounting purposes.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe)# radius accounting set1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> concurrent-requests 0-4000</td>
<td>Sets the maximum number of concurrent requests to the RADIUS server. The default value is 250 and the valid range is between 1 and 4000.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-acc)# concurrent-requests 34</td>
<td></td>
</tr>
</tbody>
</table>
### How to Implement Billing

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 7 retry-interval range</td>
<td>Sets the interval for resending an accounting request to the RADIUS server. The default value is 1200 ms and the valid range is between 10 and 10,000 ms.</td>
</tr>
</tbody>
</table>

**Example:**

```
Router(config-sbc-sbe-acc)# retry-interval 2000
```

| Step 8 retry-limit range | Sets the retry interval to the RADIUS server. The default value is 5 and the valid range is between 0 and 9. |

**Example:**

```
Router(config-sbc-sbe-acc)# retry-limit 4
```

| Step 9 server server-name | Enters the mode for configuring an accounting server within this client. |

**Example:**

```
Router(config-sbc-sbe-acc)# server Cisco-AR1-PC
```

| Step 10 address ipv4 A.B.C.D | Configures the address of an accounting server. |

**Example:**

```
Router(config-sbc-sbe-acc-ser)# address ipv4 200.200.200.153
```

| Step 11 priority pri | Configures the priority of the accounting server. The pri argument must be in the range of 1 to 10 (highest to lowest). |

**Example:**

```
Router(config-sbc-sbe-acc-ser)# priority 2
```

| Step 12 key key | Configures the RADIUS authentication key or shared secret to be used for this accounting server. |

**Example:**

```
Router(config-sbc-sbe-acc-ser)# key cisco
```

| Step 13 port port-number | Configures the port that the RADIUS server will use to receive Access-Request or Accounting-Request packets. The default port is 1813. |

**Example:**

```
Router(config-sbc-sbe-acc-ser)# port 2009
```

| Step 14 exit | Exits the current RADIUS server mode. |

**Note** Repeat Steps 9 to 14 to create multiple RADIUS accounting servers. Only one server is primary; the rest are backup. You would repeat the following commands:

- server server-name
- address ipv4 A.B.C.D.
- priority pri
- key key
- port port-number
- exit
### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>activate</td>
<td>Activates the RADIUS server.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router/Admin(config-sbc-sbe-acc)# activate</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>exit</td>
<td>Exits the current RADIUS accounting mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router/Admin(config-sbc-sbe-acc)# exit</td>
<td></td>
</tr>
</tbody>
</table>

**Note**  Repeat steps 5 to 16 to create multiple RADIUS accounting clients. You would repeat the following commands:

- `radius accounting client-name`
- `concurrent-requests num`
- `retry-interval num`
- `retry-limit num`
- `server server-name`
- `address ipv4 A.B.C.D.`
- `priority pri`
- `key key`
- `port port-num`
- `exit`
- `activate`
- `exit`

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>billing</td>
<td>Configures billing policies.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe)# billing</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>cdr endpoint-info addressing</td>
<td>Configures billing to include endpoint addressing information.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-billing)# cdr endpoint-info addressing</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>ldr-check {HH MM}</td>
<td>Configures the time of day (local time) to run the Long Duration Check (LDR).</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-billing)# ldr-check 22 30</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>local-address ipv4 {A.B.C.D.}</td>
<td>Configures the local IPv4 address that appears in the CDR.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-billing)# local-address ipv4 10.20.1.1</td>
<td></td>
</tr>
</tbody>
</table>
### How to Implement Billing

**Step 21** method packetcable-em

Example:

```
Router(config-sbc-sbe-billing)# method packetcable-em
```

Enables the packet-cable billing method.

**Step 22** cache [path {WORD} | alarm {critical VAL} | major VAL} | minor VAL} | max-size {0-4194303}]

Example:

```
Router(config-sbc-sbe-billing)# cache path harddisk:
```

Configures call detail record caching parameters, including alarm levels, maximum cache size, and cache path location.

**Note** See Tip after the table for configuring the cache path to a hard disk.

**Step 23** packetcable-em method-index transport radius RADIUS-client-name

Example:

```
Router(config-sbc-sbe-billing)# packetcable-em 4 transport radius set1
```

Configures a packet-cable billing instance.

RADIUS-client-name should match the client-name configured with the `radius accounting client-name` command.

**Step 24** batch-size number

Example:

```
Router(config-sbc-sbe-billing-packetcable-em)# batch-size 256
```

Configures the maximum size of a batch when the batch must be set immediately.

**Step 25** batch-time number

Example:

```
Router(config-sbc-sbe-billing-packetcable-em)# batch-time 22
```

Configures the maximum number of milliseconds for which any record is held before the batch is sent.

**Step 26** attach

Example:

```
Router(config-sbc-sbe-billing-packetcable-em)# attach
```

Activates the billing for a RADIUS client.

**Step 27** exit

Example:

```
Router(config-sbc-sbe-billing-packetcable-em)# exit
```

Exits the current mode.

**Note** Repeat steps 22 to 25 to create multiple billing method instances. You would repeat the following commands:

- `packetcable-em method-index transport radius RADIUS-client-name`
- `batch-size number`
- `batch-time number`
- `attach`

**Step 28** activate

Example:

```
Router(config-sbc-sbe-billing)# activate
```

Activates the Billing Manager.
If you choose to set the cache path to hard disk, the cache files are created in the root directory. To prevent cluttering up your root directory, we recommend the following steps:

1. Make a directory on the disk to store billing records. For example: mkdir harddisk:billcache
2. Configure the cache path to point to this directory. For example, the following command configures the cache path to point to the directory billcache:
   ```bash
cache path harddisk:/billcache/
```

**Tip**

```bash
tip The trailing forward slash / is mandatory in the cache path configuration.
```

### Configuration Examples of Implementing Billing

The following example configures billing and enables caching of call detail records and event messages on the designated hard disk:

```bash
Router# configure terminal
Router(config)# sbc mysbc
Router(config-sbc)# sbe
Router(config-sbc-sbe)# control address aaa ipv4 10.10.10.1 vrf default
Router(config-sbc-sbe)# radius accounting mars
Router(config-sbc-sbe-acc)# concurrent-requests 300
Router(config-sbc-sbe-acc)# retry-interval 1000
Router(config-sbc-sbe-acc)# retry-limit 6
Router(config-sbc-sbe-acc)# server moon
Router(config-sbc-sbe-acc)# address ipv4 10.20.1.1
Router(config-sbc-sbe-acc)# priority 4
Router(config-sbc-sbe-acc)# key test
Router(config-sbc-sbe-acc)# port 1820
Router(config-sbc-sbe-acc)# exit
Router(config-sbc-sbe-acc)# activate
Router(config-sbc-sbe-acc)# exit
Router(config-sbc-sbe)# billing
Router(config-sbc-sbe-billing)# ldr-check 22 30
Router(config-sbc-sbe-billing)# local-address ipv4 10.20.1.1
Router(config-sbc-sbe-billing)# method packetcable-em
Router(config-sbc-sbe-billing)# cache path harddisk:
Router(config-sbc-sbe-billing)# packetcable-em 3 transport radius test
Router(config-sbc-sbe-billing-packetcable-em)# batch-size 256
Router(config-sbc-sbe-billing-packetcable-em)# batch-time 22
Router(config-sbc-sbe-billing-packetcable-em)# attach
Router(config-sbc-sbe-billing-packetcable-em)# exit
Router(config-sbc-sbe-billing)# activate
```

The following configuration example shows that cache is enabled on the hard disk:

```bash
sbc asr
sbe
  ! - Local radius IP address
  control address aaa ipv4 10.1.1.1

  ! - First radius accounting client group
  radius accounting ACCT-CLIENT-GROUP-1
  ! - First radius server
  server ACCT-SERVER-1
  address ipv4 20.1.1.1
```
The following configuration example shows that four RADIUS servers have been configured in pairs; the second RADIUS server is backing up server 1, the third RADIUS server is backing up server 4, and both pairs of servers are receiving copies of the same records:

```
sbc asr
sbe
    ! - Local radius IP address
    control address aaa ipv4 10.1.1.1

    ! - First radius accounting client group
    radius accounting ACCT-CLIENT-GROUP-1
    ! - First radius server
    server ACCT-SERVER-1
    address ipv4 20.1.1.1
    key cisco
    ! - Backup for First radius server
    server ACCT-SERVER-2
    address ipv4 20.1.1.2
    key cisco
    activate

    ! - Second radius accounting client group
    radius accounting ACCT-CLIENT-GROUP-2
    ! - Second radius server
    server ACCT-SERVER-3
    address ipv4 30.1.1.1
    key cisco
    ! - Backup for Second radius server
    server ACCT-SERVER-4
    address ipv4 30.1.1.2
    key cisco
    activate

    ! - Billing Manager.
    billing
    local-address ipv4 10.1.1.1
    method packetcable-em
    cache path harddisk:
    ! - First billing method.
    packetcable-em 0 transport radius ACCT-CLIENT-GROUP-1
    local-address ipv4 10.1.1.1
    attach
    activate

    ! - Second billing method for duplicate records.
    packetcable-em 1 transport radius ACCT-CLIENT-GROUP-2
    local-address ipv4 10.1.1.1
    attach
    activate
```
The following configuration example shows how to configure endpoint information to capture address information:

```
Router# configure terminal
Router(config)# sbc mySBC
Router(config-sbc)# sbe
Router(config-sbc-sbe)# billing
Router(config-sbc-sbe-billing)# cdr endpoint-info addressing
Router(config-sbc-sbe-billing)# end
Router#
```

The following show command output shows that the billing is configured to include the addressing information of the endpoint:

```
Router# show sbc mySBC sbe billing instance
Billing Manager Information:
  Local IP address:            172.18.53.179
  LDR check time:              0 :0
  Method                       packetcable-em
  Method                       packetcable-li
  Admin Status:                DOWN
  Operation Status:            DOWN
  Cache path:                  usb0:billing_cache/
  Cache max size:              0 Kilobytes
  Cache minor-alarm:           97656 Kilobytes
  Cache major-alarm:           488281 Kilobytes
  Cache critical-alarm:        976562 Kilobytes
  Retry-interval:              20 secs
  CDR Media-Info:              Not Included
  CDR Endpoint-Info:           Addressing
Billing Methods:
  Radius client name:          ssss
  Instance:                    0
  Type:                        PACKET-CABLE
  Transport Mechanism Status:  DOWN
  Active Calls Billed:         0
  Local IP Address:            172.18.53.179
  Deact-mode:                  abort
  Admin Status:                DOWN
  Operation Status:            DOWN
  LDR check time:              0 :0
  Batch size:                  0
  Batch time:                  1000 ms
```

**Support Billing for IP Format**

Internet is no longer used to transmit only data; it is also used to transmit voice and video. Although the transmission of voice and video through Internet has simplified communication to a large extent, it is very important to understand how voice and video services are being managed and configured.

The PacketCable billing method that is being currently used by the SBC generates call detail record (CDR) in the Bellcore AMA Format (BAF). However, the BAF format is too telephony-specific, and does not contain sufficient provision to support IP-centric logging information. For example, the BAF format does not record session description protocol (SDP) or real-time transport control protocol (RTCP) statistics. Moreover, the PacketCable billing method is not extensible, because of which it is not possible to define extensions to contain these fields.
The XML-based billing method has been selected because it can process IP-centric logging information. It is flexible, and it is commonly used in situations where data must be translated between different platforms, for example, translating billing data from the SBC and the billing server.

Overview of XML-Based Billing

The XML-based billing method is used to generate a set of XML records, each of which gives a complete description of a call. For each call, there is an XML record. In the XM billing method, the billing events are generated and stored in the Billing Manager. Only after the call is complete, the Billing Manager writes the complete CDR on the disk. The XML billing method stores the billing records using the local file daemon. The XML billing records are stored locally in the path configured using the command-line interface (CLI).

When a call begins, the SBC starts recording the billable events pertaining to that call. After the call is completed, the SBC stops recording, and collates the events into a single CDR. The format of the CDR is a proprietary XML format, which can be analyzed and post-processed with standard XML parser tools. The CDR is appended to a local file. Critical, major, and minor alarms for notifying administrator for increase in file-size upon exceeding the configured threshold value is configured using the `cdr alarm` command. This enables the administrator to free up disk space before the disk gets full and the old billing information gets overwritten by the new billing information.

For more information on XML billing schema, see Appendix 1, ?$paratext>?.

Restrictions for XML-Based Billing

Following are restrictions for XML-based Billing:

- A maximum of only one XML billing instance can be configured.
- Each billing method configuration (under billing) consumes memory. A billing method should not be configured, unless at least one instance of the corresponding method is also configured.
- The `no method xml` command fails if an instance of the corresponding method is configured.
- Compression of the billing records is not supported.
- H.323 is supported for XML billing, but with some limitations. One such limitation is that no H.323 signalling address is present in XML billing instances.

Configuring XML-Based Billing

The following section provides configuration details for the XML billing method, the local path to store the CDR records, threshold values, and for configuring other parameters:

**SUMMARY STEPS**

1. `configure terminal`
2. `sbc sbcname`
3. `sbe`
4. `billing`
5. `method xml`
6. **xml** `xmlinstance`  
7. **cdr path** `path`  
8. **ldr-check** `hour:min`  
9. **cdr alarm minor** `2000` **major** `1000` **critical** `500`  
10. **flipped-interval** `240`  
11. **flipped-size** `20480`  
12. **deact-mode** `quiesce`  
13. **attach**  
14. **exit**  
15. **activate**

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>configure terminal</strong> Enters the global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>configure terminal</code></td>
</tr>
<tr>
<td></td>
<td><strong>Step 2</strong> <code>sbc service-name</code> Enters the mode of an SBC service.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> <code>sbc mysbc</code> Use the <code>service-name</code> argument to define the name of the service.</td>
</tr>
<tr>
<td></td>
<td><strong>Step 3</strong> <code>sbe</code> Enters the mode of an SBE entity within an SBC service.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> <code>sbe</code></td>
</tr>
<tr>
<td></td>
<td><strong>Step 4</strong> <code>billing</code> Configures billing policies.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> <code>billing</code> <strong>Note</strong> There can be only one instance of Billing Manager per SBC. The Billing Manager must be configured to configure billing.</td>
</tr>
<tr>
<td></td>
<td><strong>Step 5</strong> <code>method xml</code> Enables the XML billing method.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> <code>method xml</code></td>
</tr>
<tr>
<td></td>
<td><strong>Step 6</strong> <code>xml method-instance</code> Configures an XML billing instance. The range of valid values are 0 to 7.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> <code>xml</code> <code>xml 1</code></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 7</strong> <code>cdr path path</code></td>
<td>Configures the path to store the CDR billing records. The path must locally point to a directory located either on the flash disk or the hard drive on the Cisco ASR 1000 Series Router. Example: &lt;br/&gt;Router(config-sbc-sbe-billing-xml)# cdr path usb0:cdr</td>
</tr>
<tr>
<td><strong>Step 8</strong> <code>ldr-check hour minutes</code></td>
<td>Configures the time for checking long duration records. This is the time when all calls over 24-hours-long are reported. Example: &lt;br/&gt;Router(config-sbc-sbe-billing-xml)# ldr-check 23 30</td>
</tr>
<tr>
<td><strong>Step 9</strong> <code>cdr alarm minor 2000 major 1000 critical 500</code></td>
<td>Configures the alarms to be triggered when free disk space that is lower than the configured size is available. Example: &lt;br/&gt;Router(config-sbc-sbe-billing-xml)# cdr alarm minor 2000 major 1000 critical 500</td>
</tr>
<tr>
<td><strong>Step 10</strong> <code>flipped-interval 240</code></td>
<td>Configures the maximum interval (in seconds) to flip the billing XML file. The default value is 3 minutes. Example: &lt;br/&gt;Router(config-sbc-sbe-billing-xml)# flipped-interval 240</td>
</tr>
<tr>
<td><strong>Step 11</strong> <code>flipped-size 20480</code></td>
<td>Configures the maximum size (in kilobytes) to flip the billing XML file. The default value is 10240 kilo bytes (KB). Example: &lt;br/&gt;Router(config-sbc-sbe-billing-xml)# flipped-size 20480</td>
</tr>
<tr>
<td><strong>Step 12</strong> <code>deact-mode quiesce</code></td>
<td>Configures the deactivate mode for the XML billing method. Example: &lt;br/&gt;Router(config-sbc-sbe-billing-xml)# deact-mode quiesce</td>
</tr>
<tr>
<td><strong>Step 13</strong> <code>attach</code></td>
<td>Activates the billing instance for XML. Example: &lt;br/&gt;Router(config-sbc-sbe-billing-xml)# attach</td>
</tr>
<tr>
<td><strong>Step 14</strong> <code>exit</code></td>
<td>Exits the current mode. Example: &lt;br/&gt;Router(config-sbc-sbe-billing-xml)# exit</td>
</tr>
<tr>
<td><strong>Step 15</strong> <code>activate</code></td>
<td>Activates the Billing Manager. Example: &lt;br/&gt;Router(config-sbc-sbe-billing)# activate</td>
</tr>
</tbody>
</table>
Retrieving the XML Billing Records

Because the CDR billing records are stored locally on the Cisco ASR 1000 Series Router, it is recommended that the XML billing records are copied to another system regularly. The SBC stores the XML file under the CDR path configured using the CLI. The XML file is flipped after exceeding the fixed size or interval configured. The default file size is 10 MB, and, the default interval is 3 minutes. Copying the billing records from the local disk to remote machine everyday and removing the old billing records from the local disk is therefore recommended. For security reasons, the file should be copied using a secure transport method such as SCP or HTTPS.

Managing Disk Space Through Alarms

The XML billing CDR records are stored on the disk by the file daemon. If there are too many calls in the system can quickly fill a disk. It is therefore important to put an automated management system in place to ensure that sufficient disk space is permanently available. An automated system uses file transfer protocol (FTP) to regularly copy the CDR files to an appropriate server, and deletes the files from the local disk.

If free disk space is lower than what is configured, the SBC generates an alarm, requesting the administrator to free up the disk space by removing the CDRs. The SBC continues to accept calls until more disk space is available. To prevent unbilling of active calls due to lack of disk space, it is recommended that minor, major, and critical alarms to be configured regularly notify the administrator to free up disk space when the free disk space threshold size is exceeded.

Managing the Billing Records During RP Failover

It is important to consider various scenarios that might need attention to retain the billing records. One such scenario is managing the billing records during route processor (RP) failover from active to standby. The SBC billing architecture is designed such that billing records are not lost in case of failover to standby RP. The architecture makes certain assumptions on the infrastructure, and those assumptions should be implemented and verified.

The Billing Manager generates transient billing control block, with billing data. The primary SBC replicates these blocks to the standby SBC. In case of a failover, the call state and billing state are available on the standby, and are designed to continue the call and bill it.

In XML-based billing, before the failover, the Billing Manager stores the XML billing records in the local disk (via the file daemon interface). When failover occurs, the file daemon flushes the billing records in cache buffer into hard disk. The file daemon writes the records in the local disk belonging to the new active RP.

Note

The CDR path for storing the XML billing records must be defined earlier on the new, active RP. If the CDR path is not defined, the billing records will not be written to the hard disk. If the CDR path is not defined, create it by executing the `cdr path path` command from the `config-sbc-sbe-billing-xml` command mode.

The old billing records that are present on the new standby RP can be copied to a remote machine using the `copy stby-harddisk: <destination path>` command.
MD5 Checksum Support for XML Billing Records File

The XML billing records that are stored locally are copied to a remote machine. To ensure that the billing records copied to the destination remote machine are the same as the one existing locally, MD5 checksum support has been implemented on the XML billing records file. A checksum is a form of mechanism that ensures that the file is downloaded properly. The MD5 checksum support is used to provide the XML billing record file integrity check, when the XML billing record is copied from a local storage to a remote server.

When an old XML billing record file is closed, SBC computes and generates the MD5 checksum for the old XML billing record file. The checksum value is stored in the MD5 checksum log file. If size of the log file is more than 2 MB, the MD5 value is switched to another log file to write. There are two log files, md5checksum1.log and md5checksum2.log. The log files are located under the CDR path configured under the SBC SBE billing XML instance.

Selective RADIUS Billing

The billing methods supported by the SBC are:

- XML billing—Billing records are written in a proprietary XML format to disk.
- PacketCable billing—RADIUS messages are sent to RADIUS servers.

Prior to Cisco IOS XE Release 3.3S, all calls have the billing records generated for all the active billing methods. However, the customer that has a RADIUS server of limited capacity cannot generate billing records for calls for a subset of all adjacencies. From Cisco IOS XE Release 3.3S, the Selective RADIUS Billing feature provides the function to select billing methods for calls relating to different adjacencies.

The billing method or methods used for calls can be selected at a per-adjacency scope and the user can also choose to not use the billing method for a specific adjacency.

Configuring Selective RADIUS Billing

This task configures the Selective RADIUS Billing feature.

SUMMARY STEPS

1. configure terminal
2. sbc sbc-name
3. sbe
4. cac-policy-set policy-set-id
5. cac-table table-name
6. table-type policy-set
7. entry entry-id
8. billing filter { enable | disable }
9. billing methods { xml | packetcable-em }
10. end
11. show sbc sbc-name sbe cac-policy-set id table name entry id
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>configure terminal</code></td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Router# configure terminal</td>
</tr>
<tr>
<td>Step 2</td>
<td><code>sbc sbc-name</code></td>
<td>Creates the SBC service on the SBC, and enters the SBC configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Router(config)# sbc mySBC</td>
</tr>
<tr>
<td>Step 3</td>
<td><code>sbe</code></td>
<td>Enters the signaling border element (SBE) function mode of the SBC.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Router(config-sbc)# sbe</td>
</tr>
<tr>
<td>Step 4</td>
<td><code>cac-policy-set policy-set-id</code></td>
<td>Enters the CAC policy set configuration mode within an SBE entity, creating a new policy set, if necessary:</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Router(config-sbc-sbe)# cac-policy-set 1</td>
</tr>
<tr>
<td>Step 5</td>
<td><code>cac-table table-name</code></td>
<td>Enters the CAC table mode for configuration of an admission control table (creating one, if necessary) within the context of an SBE policy set.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Router(config-sbc-sbe-cacpolicy)# cac-table t1</td>
</tr>
<tr>
<td>Step 6</td>
<td>`table-type (policy-set</td>
<td>limit (list of limit tables)}`</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Router(config-sbc-sbe-cacpolicy-cactable)# table-type policy-set</td>
</tr>
<tr>
<td>Step 7</td>
<td><code>entry entry-id</code></td>
<td>Enters the CAC table entry mode to modify an entry in an admission control table.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Router(config-sbc-sbe-cacpolicy-cactable)# entry 1</td>
</tr>
<tr>
<td>Step 8</td>
<td>`billing filter (enable</td>
<td>disable)`</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Router(config-sbc-sbe-cacpolicy-cactable-entry)# billing filter enable</td>
</tr>
<tr>
<td>Step 9</td>
<td>`billing methods (xml</td>
<td>packetcable-em)`</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Router(config-sbc-sbe-cacpolicy-cactable-entry)# billing methods xml</td>
</tr>
</tbody>
</table>
Support Billing for IP Format

Chapter 36 Implementing Billing on Cisco Unified Border Element (SP Edition)

Command or Action | Purpose
--- | ---
Step 10 end | Enables exit from the CAC table entry configuration mode and entry into the Privileged EXEC mode.

Example:
Router# end

Step 11 show sbc sbc-name sbe cac-policy-set id table name entry id | Lists the detailed information for a given entry in a CAC policy table.

Example:
Router# show sbc mySBC sbe cac-policy-set 1 table t1 entry 1

The following example displays the partial output of the show sbc sbc-name sbe cac-policy-set id table name entry id command that lists the billing filter information:

Router# show sbc mySBC sbe cac-policy-set 1 table t1 entry 1

SBC Service "mySBC"
CAC Averaging period 1: 60 sec
CAC Averaging period 2: 0 sec

CAC Policy Set 1
Active policy set: No
Description:
First CAC table:
First CAC scope: global

Table name: t1
Description:
Table type: policy-set
Total call setup failures (due to non-media limits): 0

Entry 1
CAC scope:
CAC scope prefix length: 0
Action: Not set
Number of call setup failures (due to non-media limits): 0

media bandwidth policing: Degrade
Media policy limit: mp1
IPsec maximum registers: 10
IPsec maximum calls: 5
Billing filter: enable
Billing filter methods: xml

Configuration Example of Selective RADIUS Billing

The following example configures all calls billed using XML billing, all calls on an adjacency in the IMS-adjacencies group are configured to be billed using XML and PacketCable-em billing, however, all calls on a special-adj adjacency are configured for not being billed at all.

cac-policy-set 1
first-cac-scope global
first-cac-table 1
table-type limit adj-group
cac-table 1
entry 1
  action next-table 2
  billing filter enable
  billing methods xml
!
!
cac-table 2
entry 1
  match-value ims-adjacencies
  action next-table 3
  billing filter enable
  billing methods xml
  billing methods packetcable-em
!
!
cac-table 3
entry 1
  match-value special-adj
  action cac-complete
  billing filter enable
!
!
!
Billing Support

The following sections describe billing and its many aspects. It is critical to understand all Cisco Unified Border Element (SP Edition) billing features and capabilities before performing billing configurations.

- Integrated Billing Systems, page 37-1
- Event Message Transmission, page 37-3
- Supported Event Message Detail, page 37-6
- Administration and Configuration, page 37-13
- Logging and Alarms, page 37-13
- Fault Tolerance, page 37-14
- Example for Event Messages from Cisco Unified Border Element (SP Edition) to RADIUS Billing Server, page 37-14
- Security, page 37-39

Integrated Billing Systems

Integrated billing is achieved through the PacketCable Event Messages architecture (see the PacketCable 1.5 Event Messages Specification; PKT-SP-EM1.5-101-050128) as exemplified in Figure 37-1 where Cisco Unified Border Element (SP Edition) is integrated into this architecture. As shown, the billing server supports PacketCable Event Messages.

Cisco Unified Border Element (SP Edition) on the Cisco ASR 1000 Series Routers supports remote billing in the unified mode. Remote billing is call billing that is integrated with a third-party accounting server.
Figure 37-1 shows Cisco Unified Border Element (SP Edition) operating in a unified model where the billing system is being deployed with three billing servers. Cisco Unified Border Element (SP Edition) can be configured to send to these servers in a range of ways, such as to all three simultaneously, or to use one primary and two backups.

**Figure 37-1  Integrated Billing Deployment**

The system operates as follows:

- Cisco Unified Border Element (SP Edition) produces event messages (EMs). These event messages are for billable or other interesting events, such as call start, call end, and media-type changes.
- Cisco Unified Border Element (SP Edition) and other elements of the system, which produces EMs, sends them in real time (or batched up for network efficiency) using the RADIUS protocol to the billing server.

**Note**

The *PacketCable 1.5 Event Messages Specification* discusses sending the identifying information (the BCID and FEID) on the outgoing INVITE and responding SDP so that correlation can be done between the two sets of billing data. Cisco Unified Border Element (SP Edition) does not support this mechanism for intra-domain or inter-domain transmission. The billing server must perform the correlation using an alternative method (for example, using the telephone numbers dialed and the time of the call).
Event Message Transmission

The generated event messages, as described in the Event Messages Set Overview section, are sent using the RADIUS protocol to a preconfigured set of billing servers. Before getting into the actual detail of the event messages, review the event message transmission considerations described in the following sections:

- Multiple Server Support
- Event Message Batching
- Event Messages Set Overview

Multiple Server Support

Billing servers are configured at start-up, in SETs:

- Each SET contains a list of one or more billing servers, consisting of a single primary server and an ordered list of zero or more backup servers.
- The SBE can be configured with one or more sets of billing servers.

Each event message is sent to the entire collection of sets, but to only one machine within each set.

- For each set, the SBE sends the event message to the primary server within the set.
- If the primary server is unavailable, the message is sent to the first backup server (if present). If the first backup server is also unavailable, the message is sent to the second backup and so on until either a machine accepts the message or all the servers in the set have been tried.
- If there are no machines in a set accepting messages, the entire set is marked as unavailable.

Figure 37-2 shows the multiple server support.
Event Message Batching

Because of the inefficiency of the RADIUS protocol, the SBE collates event messages into batches and sends them using a single RADIUS message to alleviate the burden on the transport mechanism.

Batching is possible only on a per-set basis. The batch size is not configurable, but is determined by the load on the billing component.

It is not possible to disable batching.

Event Messages Set Overview

This section specifies the set of event messages supported by Cisco Unified Border Element (SP Edition):

- Call-Specific Messages, page 37-5
- Out-of-Band Messages, page 37-5
- Unsupported Messages, page 37-5
Call-Specific Messages

The following table lists supported call event messages.

<table>
<thead>
<tr>
<th>Event Message</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signaling_Start</td>
<td>Sent when signaling has begun (inbound) and when it is about to begin (outbound); for example, received INVITE on inbound and about to send INVITE on outbound for a SIP endpoint</td>
</tr>
<tr>
<td>QoS_Resolve</td>
<td>Sent when there is reserved QoS in the DBE. Sent for the inbound leg when the inbound QoS is reserved, and for the outbound leg when we reserve the outbound QoS is reserved.</td>
</tr>
<tr>
<td>Call_Answer</td>
<td>Indicates that the terminating party has answered and that media has started. This message is sent for both legs at the same time.</td>
</tr>
<tr>
<td>QoS_Commit</td>
<td>Sent when QoS is committed by the DBE. This message is sent for both legs at the same time.</td>
</tr>
<tr>
<td>Call_Disconnect</td>
<td>Sent when the call has been terminated and the media has ceased flowing. Sent for both legs at the same time.</td>
</tr>
<tr>
<td>QoS_Release</td>
<td>Sent when the QoS has been released by the DBE. Sent for both legs at the same time.</td>
</tr>
<tr>
<td>Signaling_Stop</td>
<td>Sent after all signaling is complete for each party in the call. (The event is generated once for each party, when the last signaling message has been sent.)</td>
</tr>
<tr>
<td>Media_Statistics</td>
<td>Media statistics for the call as reported by the DBE. This is sent for each leg when the media is released.</td>
</tr>
<tr>
<td>Media_Alive</td>
<td>Indicates that a long-duration call is still active. This is sent for each leg of the call, at a preconfigured time of day, every 24 hours.</td>
</tr>
</tbody>
</table>

Out-of-Band Messages

The following table lists event messages that are non-call-related, out-of-band event messages.

<table>
<thead>
<tr>
<th>Event Message</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time_Change</td>
<td>Sent when changes of more than 200 ms occur in the time; also sent for daylight savings changes, and so on.</td>
</tr>
</tbody>
</table>

Unsupported Messages

The following table lists the event messages that are not supported.

<table>
<thead>
<tr>
<th>Event Message</th>
<th>Notes</th>
<th>Why Not Supported?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database_Query</td>
<td>Sent when querying external databases about toll-free carriers, LNP routing, and so on.</td>
<td>Cisco Unified Border Element (SP Edition) does not support database queries.</td>
</tr>
</tbody>
</table>
Supported Event Message Detail

This section specifies the supported event messages and the attributes sent for each one.

Signaling_Start

This message is sent when signaling starts for a call; that is, when Cisco Unified Border Element (SP Edition) has ascertained that the destination is routable and the originating endpoint is allowed to make the call (that is, after the SLA has been checked).

The following table lists the attributes sent with this message.

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>EM_Header</td>
<td>Common header attribute.</td>
</tr>
<tr>
<td>Direction_Indicator</td>
<td>Specifies if the device represents an originating or terminating part of the call.</td>
</tr>
<tr>
<td></td>
<td>1 = originating</td>
</tr>
<tr>
<td></td>
<td>2 = terminating</td>
</tr>
<tr>
<td>MTA_Endpoint_Name</td>
<td>The string <em>MTA Endpoint</em> or the source endpoint information (adjacency name or addressing information).</td>
</tr>
<tr>
<td></td>
<td>The value of this field is either set to <em>MTA Endpoint</em> or to the endpoint information.</td>
</tr>
<tr>
<td></td>
<td>The source adjacency name is used if the SBC is configured to include the adjacency name in the billing records and if the message is from the originating device.</td>
</tr>
<tr>
<td></td>
<td>The source addressing information—in the format <em>IP address,port,transport type, adjacency name</em>—is used if the SBC is configured to include the addressing information in the billing records and if the message is from the originating device.</td>
</tr>
<tr>
<td>Calling_Party_Number</td>
<td>The number of the calling party (if available).</td>
</tr>
</tbody>
</table>
Supported Event Message Detail

The following table lists the attributes not sent with this message.

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Called_Party_Number</td>
<td>The number of the called party (always present).</td>
</tr>
<tr>
<td>Routing_Number</td>
<td>Indicates a routable number (always present).</td>
</tr>
<tr>
<td>Billing_Type</td>
<td>Included when the originating endpoint is a measured rate subscriber.</td>
</tr>
</tbody>
</table>

The following table lists the attributes sent with this message.

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Called_Party_Number</td>
<td>The number of the called party (always present).</td>
</tr>
<tr>
<td>Routing_Number</td>
<td>Indicates a routable number (always present).</td>
</tr>
<tr>
<td>Billing_Type</td>
<td>Included when the originating endpoint is a measured rate subscriber.</td>
</tr>
<tr>
<td>Location_Routing_Number</td>
<td>LNP not supported.</td>
</tr>
<tr>
<td>Carrier_Identification_Code</td>
<td>PSTN interfacing not supported (softswitch function).</td>
</tr>
<tr>
<td>Trunk_Group_ID</td>
<td>As above.</td>
</tr>
<tr>
<td>Intl_Code</td>
<td>Indicates the origin of an international call.</td>
</tr>
<tr>
<td>Dial_Around_Code</td>
<td>Carrier specification via dial-around codes not supported.</td>
</tr>
<tr>
<td>Jurisdiction_Information_Parameter</td>
<td>Ported-In billing not supported [transparent to Cisco Unified Border Element (SP Edition)].</td>
</tr>
<tr>
<td>Ported_In_Calling_Number</td>
<td>As above.</td>
</tr>
<tr>
<td>Ported_In_Called_Number</td>
<td>As above.</td>
</tr>
<tr>
<td>Called_Party_NP_source</td>
<td>LNP not supported.</td>
</tr>
<tr>
<td>Calling_Party_NP_source</td>
<td>As above.</td>
</tr>
</tbody>
</table>

**QoS_Reserve**

This message is generated when the SBE has reserved bandwidth (QoS) on the network through the DBE.

If this reserved bandwidth changes, this message (along with the partner QoS_Commit message) is generated anew.

If the SBE is managing multiple gates, this message is generated only for the gates to and from each MTA endpoint (and not the internal gates). There are no optional attributes not sent on this message.

The following table lists the attributes sent with this message.

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>EM_Header</td>
<td>Common header attribute.</td>
</tr>
<tr>
<td>QoS_Descriptor</td>
<td>Description of the QoS reserved (see below).</td>
</tr>
<tr>
<td>MTA_UDP_Portnum</td>
<td>The UDP port number on the network element endpoint.</td>
</tr>
<tr>
<td>Flow_Direction</td>
<td>1 = upstream</td>
</tr>
<tr>
<td></td>
<td>2 = downstream</td>
</tr>
<tr>
<td>SF_ID</td>
<td>This is a required, DOCSIS-specific attribute, generated by the CMTS in a PacketCable architecture. Because Cisco Unified Border Element (SP Edition) does not support DOCSIS, this attribute is always 0.</td>
</tr>
</tbody>
</table>
Call_Answer

This message indicates the earliest point at which non-early two-way media is established.

The SBE sends the message to the billing servers when it is notified that the called party has gone off-hook; that is, that they have answered the call.

The following table lists the attributes sent with this message.

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>EM_Header</td>
<td>Common header attribute</td>
</tr>
<tr>
<td>Charge_Number</td>
<td>The charge number in the appropriate cases such as collect call, calling-card call, call billed to a third party, or others. For Cisco Unified Border Element (SP Edition), this number is always the calling number.</td>
</tr>
<tr>
<td>MTA_Endpoint_Name</td>
<td>The destination endpoint information—adjacency name or addressing information—is added to the message if the SBC is configured to include the endpoint information in the billing records and if the SBC is the terminating device. If the SBC is not configured to include the endpoint information in the message, this attribute is not included.</td>
</tr>
</tbody>
</table>

The following table lists the attributes not sent with this message.

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Related_Call_Billing_Correlation_ID</td>
<td>The BCID assigned to the leg from the terminating network element. Cisco Unified Border Element (SP Edition) does not share BCID and FEID information with other network elements.</td>
</tr>
<tr>
<td>FEID</td>
<td>Contains the FEID assigned to the network element at the other end of the leg. Cisco Unified Border Element (SP Edition) does not share BCID and FEID information with other network elements.</td>
</tr>
</tbody>
</table>

QoS_Commit

This message is sent by the SBE when the gate bandwidth is committed. This message is only sent if a QoS_Reservation has been previously sent.

The following table lists the attributes sent with this message.

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>EM_Header</td>
<td>Common header attribute</td>
</tr>
<tr>
<td>MTA_UDP_Portnum</td>
<td>The UDP port number on the network element endpoint.</td>
</tr>
<tr>
<td>Flow_Direction</td>
<td>1 = upstream</td>
</tr>
<tr>
<td></td>
<td>2 = downstream</td>
</tr>
</tbody>
</table>
### SF_ID
- Always 0 (Cisco Unified Border Element (SP Edition) does not support DOCSIS).

### Total_Bandwidth (attribute ID 253)
- The total bandwidth in use by the streams described in this QoS_Commit message. See Table B-11 for the structure of this attribute.

### Media_Session_Desc (attribute ID 254)
- Zero or more attributes describing the media committed in this Flow Direction. If more than one flow is committed, multiple Media_Session_Desc attributes are differentiated by the Stream_IDs. See Table B-12 for the structure of this attribute.

#### Attribute Name | Comment
--- | ---
QoS_Descriptor | Information is sent on the QoS_Reserve message and not duplicated on this message.

The following table lists the attributes not sent with this message.

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Length</th>
<th>Type</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total_Bandwidth (attribute ID 253)</td>
<td>8</td>
<td>unsigned integer</td>
<td>The total bandwidth in use by the streams described in this QoS_Commit message.</td>
</tr>
</tbody>
</table>

The following table lists the structure of the Total_Bandwidth attribute (attribute ID 253).

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Length</th>
<th>Type</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stream_ID</td>
<td>4</td>
<td>unsigned integer</td>
<td>Unique stream identifier within the scope of the call. A Stream_ID identifies two flows, one upstream and one downstream - this Media_Info attribute is for the flow identified by the Flow_Direction attribute on this message.</td>
</tr>
<tr>
<td>Local_address_type</td>
<td>1</td>
<td>Byte</td>
<td>1 = IPv4 address—of length 4. 2 = IPv6 address—of length 16.</td>
</tr>
<tr>
<td>Local_address</td>
<td>variable</td>
<td>byte array</td>
<td>The local address - length defined by the Local_address_type.</td>
</tr>
<tr>
<td>Local_port</td>
<td>2</td>
<td>unsigned integer</td>
<td>The local port.</td>
</tr>
<tr>
<td>Remote_address_type</td>
<td>1</td>
<td>Byte</td>
<td>1 = IPv4 address—of length 4. 2 = IPv6 address—of length 16.</td>
</tr>
<tr>
<td>Remote_address</td>
<td>variable</td>
<td>byte array</td>
<td>The remote address—length defined by the Remote_address_type.</td>
</tr>
<tr>
<td>Remote_port</td>
<td>2</td>
<td>unsigned integer</td>
<td>The remote port.</td>
</tr>
</tbody>
</table>
Supported Event Message Detail

Chapter 37      Billing Support

Call_Disconnect

This message is generated by the SBE when 2-way media flow terminates—when sending a 200 OK response to a BYE from either party.

Usually, this message immediately precedes QoS_Release and Signaling_Stop.

This message is only sent if a Call_Answer has previously been sent.

The following table lists the attributes sent with this message.

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>EM_Header</td>
<td>Common header attribute.</td>
</tr>
<tr>
<td>Call_Termination_Cause</td>
<td>Reason for termination of the call.</td>
</tr>
<tr>
<td>There are no optional attributes not sent for this message.</td>
<td></td>
</tr>
</tbody>
</table>

QoS_Release

This message is generated by the SBE when the reserved bandwidth has been released; that is, the gate on the DBE has been closed.

The following table lists the attributes sent with this message.

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>EM_Header</td>
<td>Common header attribute.</td>
</tr>
<tr>
<td>Flow_Direction</td>
<td>1 = upstream.</td>
</tr>
<tr>
<td></td>
<td>2 = downstream.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SDP_fragment_len</th>
<th>0—The following SDP is not truncated.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1—The following SDP was truncated after the last complete a= line to prevent the RADIUS attribute from exceeding 247 bytes.</td>
</tr>
<tr>
<td>SDP_fragment</td>
<td>0—Transrating is not in use on this stream.</td>
</tr>
<tr>
<td></td>
<td>1—Transrating is in use on this stream.</td>
</tr>
<tr>
<td>Transrated</td>
<td>0—The following SDP is not truncated.</td>
</tr>
<tr>
<td></td>
<td>1—The following SDP was truncated after the last complete a= line to prevent the RADIUS attribute from exceeding 247 bytes.</td>
</tr>
<tr>
<td>Truncated</td>
<td>0—Transrating is not in use on this stream.</td>
</tr>
<tr>
<td></td>
<td>1—Transrating is in use on this stream.</td>
</tr>
<tr>
<td>Truncated</td>
<td>0—The following SDP is not truncated.</td>
</tr>
<tr>
<td>Truncated</td>
<td>1—The following SDP was truncated after the last complete a= line to prevent the RADIUS attribute from exceeding 247 bytes.</td>
</tr>
<tr>
<td>Truncated</td>
<td>0—Transrating is not in use on this stream.</td>
</tr>
<tr>
<td>Truncated</td>
<td>1—Transrating is in use on this stream.</td>
</tr>
<tr>
<td>Truncated</td>
<td>0—The following SDP is not truncated.</td>
</tr>
<tr>
<td>Truncated</td>
<td>1—The following SDP was truncated after the last complete a= line to prevent the RADIUS attribute from exceeding 247 bytes.</td>
</tr>
<tr>
<td>Truncated</td>
<td>0—Transrating is not in use on this stream.</td>
</tr>
<tr>
<td>Truncated</td>
<td>1—Transrating is in use on this stream.</td>
</tr>
</tbody>
</table>
Chapter 37      Billing Support

Supported Event Message Detail

There are no optional attributes not sent for this message.

**Signaling_Stop**

This message is sent when:

- The terminating signaling request (for example, a SIP BYE) from the party terminating the call is acknowledged by the SBE
- The terminating signaling request for the party not terminating the call is sent by the SBE and acknowledged by that party.

This message is not sent if we have not sent a Signaling_Start for this call.

The following table lists the attributes sent with this message.

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>EM_Header</td>
<td>The header attribute (must be first).</td>
</tr>
<tr>
<td>Related_Call_Billing_Correlation_ID</td>
<td>The BCID of the other leg (that is, if this is the caller, then the callee, and vice-versa).</td>
</tr>
<tr>
<td>Call_Termination_Cause</td>
<td>The reason the call was terminated.</td>
</tr>
<tr>
<td>MTA_Endpoint_Name</td>
<td>If the SBC is configured to include the endpoint information—adjacency name or addressing information—in the message, this attribute is added. The destination endpoint information is added to the terminating device message and the source endpoint information is added to the originating device message. If the SBC is not configured to include the endpoint information in the message, this attribute is not included.</td>
</tr>
</tbody>
</table>

The following table lists the attributes not sent with this message.

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>SF_ID</td>
<td>A DOCSIS specific attribute, service flow ID, generated by the CMTS in a PacketCable architecture. Cisco Unified Border Element (SP Edition) does not support DOCSIS. Therefore this attribute is always set to 0.</td>
</tr>
<tr>
<td>Media_Session_Desc (attribute ID 254)</td>
<td>Zero or more attributes describing the media committed in this Flow_Direction. If more than one flow is committed, multiple Media_Session_Desc attributes are differentiated by the Stream_IDS. See Table B-12 for the structure of this attribute.</td>
</tr>
</tbody>
</table>

There are no optional attributes not sent for this message.
Media_Statistics

When a call is terminated on the DBE (that is, the gate is closed), statistics are returned to the SBE. On receipt of these statistics, this message is generated.

When media QoS is renegotiated, the gate is closed and re-opened. In this case, statistics are logged for the first gate when it closes, and for the second gate when it closes (at the end of the call).

There may be multiple gates for each Media. The statistics are aggregated and result in only one Media_Statistics message per billing leg.

The following table lists the attributes sent with this message.

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>EM_Header</td>
<td>Common header attribute.</td>
</tr>
<tr>
<td>RTCP_Data</td>
<td>The report data from the DBE on the gate statistics.</td>
</tr>
</tbody>
</table>

There are no optional attributes not sent for this message.

Media_Alive

This message is generated once a day, at a pre-configured time.

At the preconfigured time, the SBE audits the active calls, and determines which calls (if any) have been active for more than 24 hours. For each call satisfying this condition, a Media_Alive message is generated.

The following table lists the attributes sent with this message.

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>EM_Header</td>
<td>Common header attribute.</td>
</tr>
</tbody>
</table>

There are no optional attributes not sent for this message.

Time_Change

This message is generated by the SBE either on its own behalf, or on the behalf of the DBE, when either the DBE or SBE experiences a time change of more than 200 ms (discounting slew adjustments via Network Time Protocol (NTP)). This includes step adjustments, manual time settings changes and daylight savings time adjustments.

The following table lists the attributes sent with this message.

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>EM_Header</td>
<td>Common header attribute.</td>
</tr>
<tr>
<td>Time_Adjustment</td>
<td>Adjustment in milliseconds.</td>
</tr>
</tbody>
</table>

There are no optional attributes not sent for this message.
Administration and Configuration

Billing requires the following generic configuration:

Integrated Mode Configuration

If integrated mode is specified, then the following configuration information is required:

- The assigned element ID. This is an ID assigned by the Internet service provider (ISP). The ID must be unique across the set of SBEs, sending event messages to a particular set of billing servers.
- The minor, major, and critical threshold sizes for the event message cache file.
- The location of the event message cache file on disk.
- The time at which to generate the Media_Alive message.
- RADIUS client configuration information.

Integrated mode requires the RADIUS client component of Cisco Unified Border Element (SP Edition). This has configuration requirements (such as the sets of billing servers). Each of these sets also has a state, which depends on the existence or absence of the event message cache file for that set. The administrator may change this state. The state may be disabled, active, failed, or resending.

Administering Cisco Unified Border Element (SP Edition) Billing

The billing component is administered using the Cisco Unified Border Element (SP Edition) command-line interface. See the applicable billing commands in Cisco Unified Border Element (SP Edition) Command Reference: Unified Model at:


Logging and Alarms

Alarms are tripped differently, based on how billing has been integrated, as described in the following table.

<table>
<thead>
<tr>
<th>Billing System Type</th>
<th>Logging Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integrated Billing Alarms</td>
<td>Alarms are tripped under the following conditions:</td>
</tr>
<tr>
<td></td>
<td>- Minor, major, and critical alarms are sent if the cache file size exceeds a preconfigured threshold.</td>
</tr>
<tr>
<td></td>
<td>- Alarms are tripped when billing servers become unavailable, as follows.</td>
</tr>
<tr>
<td></td>
<td>- A minor alarm is tripped if just one of the configured sets of billing servers is unavailable.</td>
</tr>
<tr>
<td></td>
<td>- A major alarm is tripped if more than one of the billing server sets is unavailable.</td>
</tr>
<tr>
<td></td>
<td>- A critical alarm is tripped if none of the billing servers is available.</td>
</tr>
<tr>
<td>Note</td>
<td>In this situation, it may be that the condition for more than one alarm is satisfied (for example, there is just one server set configured, which fails). The most severe alarm dominates.</td>
</tr>
</tbody>
</table>
Fault Tolerance

The Cisco Unified Border Element (SP Edition) billing system is fault tolerant on the following two levels:

- **Warm Failover**—Failover to a live backup (for example, a second card on the same machine).
- **Cold Failover**—Failover to a new machine with no software connection between the defunct machine and the new machine.

**Warm Failover**

During a failover to a backup system, warm failover mechanisms are supported. In the case of warm failover:

- No data is lost on the SBE.
- The value for media statistics for the call on the DBE is reset (this information is lost).

**Cold Failover**

During the failover to a cold, non-dedicated backup, some billing data is lost in the transition from the old, failed system to the new server. The number of billing records completely lost during this transition is less than 10,000 per failover. However, in such a situation, consider the following possibilities:

- The remaining billing records may be corrupted, and only partial billing records recovered. This is especially true with local CDR generation, because no logs are produced in a hard format until the call ends.
- If an event message cache exists on the failed machine, more billing events may be lost, because the disk record may be unrecoverable because of fire, hardware malfunction, or whatever the original cause of the total failure was. This, however, is an unlikely scenario, because it would require the billing server to be unavailable and unrecovered for a period preceding the cold failover.
- If the media to which the CDRs are written is lost, the entire store of CDRs not backed up (by extracting the records using FTP) is lost.
- It is not possible to detect long-duration calls following a cold failover. Data is only recoverable from the system only when an event occurs in the network, such as the media being terminated.

**Example for Event Messages from Cisco Unified Border Element (SP Edition) to RADIUS Billing Server**

This section contains the following examples:

- **Example 1** shows two requests from the SBC to the RADIUS server for a single placed call.
- **Example 2** shows requests from the SBC to RADIUS server where the SBC is configured to include the endpoint adjacency name in billing records.
- **Example 3** shows requests from the SBC to RADIUS server where the SBC is configured to include the endpoint addressing information in billing records.
Example 1

This example shows two requests from the SBC to the RADIUS server for a single placed call. The first RADIUS event message has messages related to call setup:

- Event Message Type: Signaling_Start
- Event Message Type: QoS_Reserve
- Event Message Type: Call_Answer
- Event Message Type: QoS_Commit

The second RADIUS event message has messages related to call teardown:

- Event Message Type: QoS_Release
- Event Message Type: Call_Disconnect
- Event Message Type: Signaling_Stop

Radius Protocol

Code: Accounting-Request (4)
Packet identifier: 0x0 (0)
Length: 1298
Authenticator: 25CE1B487AEB4AE70033D61B0EFS40A4A
[The response to this request is in frame 4]
Attribute Value Pairs

AVP: l=6  t=NAS-IP-Address(4): 77.111.1.51
NAS-IP-Address: 77.111.1.51 (77.111.1.51)
AVP: l=6  t=Acct-Status-Type(40): Interim-Update(3)
  Acct-Status-Type: Interim-Update (3)
AVP: l=26  t=Acct-Session-Id(44): HDq]       01+000000\000\000\000\001
  Acct-Session-Id: HDq]       01+000000
AVP: l=84  t=Vendor-Specific(26) v=CableLabs(4491)
  VSA: l=78 t=CableLabs-Event-Message(1):
    Event Message Version ID: 4
    BCID
    Timestamp: 1212445021
    Event ID: 0
    Event Counter: 1
    Event Message Type: Signaling_Start (1)
    Element Type: CMS (1)
    Element ID: 0
    Sequence Number: 0
    Event Time: 20080602221700.000
    Status: 0x00000008

(0x00000000)

(0x00000000)

(0x00000000)

Proxied (0x000000001)

VSA: l=10 t=Vendor-Specific(26) v=CableLabs(4491)
  VSA: l=14 t=CableLabs-Direction-indicator(37): Originating(1)
  CableLabs-Direction-indicator: Originating (1)
AVP: l=20  t=Vendor-Specific(26) v=CableLabs(4491)
  VSA: l=14 t=CableLabs-MTA-Endpoint-Name(3): MTA Endpoint
  CableLabs-MTA-Endpoint-Name: MTA Endpoint
Example for Event Messages from Cisco Unified Border Element (SP Edition) to RADIUS Billing Server

Chapter 37      Billing Support

Example for Event Messages from Cisco Unified Border Element (SP Edition) to RADIUS Billing Server

AVP: l=28  t=Vendor-Specific(26) v=CableLabs(4491) 123
VSA: l=22 t=CableLabs-Calling-Party-Number(4): 123
CableLabs-Calling-Party-Number: 123
AVP: l=28  t=Vendor-Specific(26) v=CableLabs(4491)
VSA: l=22 t=CableLabs-Called-Party-Number(5): service
CableLabs-Called-Party-Number: service
AVP: l=28  t=Vendor-Specific(26) v=CableLabs(4491)
VSA: l=22 t=CableLabs-Routing-Number(25): service
CableLabs-Routing-Number: service
AVP: l=10  t=Vendor-Specific(26) v=CableLabs(4491)
VSA: l=4 t=CableLabs-Billing-Type(87): 3
CableLabs-Billing-Type: 3
AVP: l=84  t=Vendor-Specific(26) v=CableLabs(4491)
VSA: l=78 t=CableLabs-Event-Message(1):
  Event Message Version ID: 4
  BCID
  Timestamp: 1212445021
  Element ID: 0
  Time Zone: DST: 1, Offset: +000000
  Event Counter: 2
  Event Message Type: Signaling_Start (1)
  Element Type: CMS (1)
  Element ID: 0
  Time Zone: DST: 1, Offset: +000000
  Sequence Number: 1
  Event Time: 20080602221700.000
  Status: 0x00000008
  .... .... .... .... .... .... .... ..00 = Status: No Error
  (0x00000000)
  .... .... .... .... .... .... .... .0.. = Event Origin: Trusted
  Element (0x00000000)
  .... .... .... .... .... .... .... 1... = Event Message Proxied:
  Proxied (0x00000001)
  Priority: 128
  Attribute Count: 6
  Event Object: 0
AVP: l=10  t=Vendor-Specific(26) v=CableLabs(4491)
VSA: l=4 t=CableLabs-Direction-indicator(37): Terminating (2)
  CableLabs-Direction-indicator: Terminating (2)
AVP: l=20  t=Vendor-Specific(26) v=CableLabs(4491)
VSA: l=14 t=CableLabs-MTA-Endpoint-Name(3): MTA Endpoint
  CableLabs-MTA-Endpoint-Name: MTA Endpoint
AVP: l=28  t=Vendor-Specific(26) v=CableLabs(4491)
VSA: l=22 t=CableLabs-Calling-Party-Number(4): 123
  CableLabs-Calling-Party-Number: 123
AVP: l=28  t=Vendor-Specific(26) v=CableLabs(4491)
VSA: l=22 t=CableLabs-Called-Party-Number(5): service
  CableLabs-Called-Party-Number: service
AVP: l=28  t=Vendor-Specific(26) v=CableLabs(4491)
VSA: l=22 t=CableLabs-Routing-Number(25): service
  CableLabs-Routing-Number: service
AVP: l=10  t=Vendor-Specific(26) v=CableLabs(4491)
VSA: l=4 t=CableLabs-Billing-Type(87): 3
  CableLabs-Billing-Type: 3
AVP: l=84  t=Vendor-Specific(26) v=CableLabs(4491)
VSA: l=78 t=CableLabs-Event-Message(1):
  Event Message Version ID: 4
  BCID
  Timestamp: 1212445021
  Element ID: 0
  Time Zone: DST: 1, Offset: +000000
  Event Counter: 1
  Event Message Type: QoS_Reserve (7)
  Element Type: CMS (1)
Example for Event Messages from Cisco Unified Border Element (SP Edition) to RADIUS Billing Server

Element ID: 0
Time Zone: DST: 1, Offset: +000000
Sequence Number: 2
Event Time: 20080602221700.000
Status: 0x00000008

(0x00000000)

Element (0x00000000)

Proxied (0x00000001)

Priority: 128
Attribute Count: 4
Event Object: 0
AVP: l=32 t=Vendor-Specific(26) v=CableLabs(4491)
VSA: l=26 t=CableLabs-QoS-Descriptor(32):
  QoS Status: 0x00000005

Service Class Name:
  Service Flow Scheduling Type: 1
AVP: l=12 t=Vendor-Specific(26) v=CableLabs(4491)
VSA: l=6 t=CableLabs-MTA-UDP-Portnum(26):
  CableLabs-MTA-UDP-Portnum: 0
AVP: l=12 t=Vendor-Specific(26) v=CableLabs(4491)
VSA: l=6 t=CableLabs-SF-ID(30):
  CableLabs-SF-ID: 0
AVP: l=10 t=Vendor-Specific(26) v=CableLabs(4491)
VSA: l=4 t=CableLabs-Flow-Direction(50):
  CableLabs-Flow-Direction: Upstream (1)
AVP: l=84 t=Vendor-Specific(26) v=CableLabs(4491)
VSA: l=78 t=CableLabs-Event-Message(1):
  Event Message Version ID: 4
BCID
  Timestamp: 1212445021
  Element ID: 0
  Time Zone: DST: 1, Offset: +000000
  Event Counter: 2
  Event Message Type: QoS_Reserve (7)
  Element Type: CMS (1)
  Element ID: 0
  Time Zone: DST: 1, Offset: +000000
  Sequence Number: 3
Example for Event Messages from Cisco Unified Border Element (SP Edition) to RADIUS Billing Server

Event Time: 20080602221700.000
Status: 0x00000008
.............. ........... ......... ....00 = Status: No Error
(0x00000000)
.............. ........... ......... ....0.. = Event Origin: Trusted
Element (0x00000000)
.............. ........... ......... ....1... = Event Message Proxied:
Proxied (0x00000001)

Priority: 128
Attribute Count: 4
Event Object: 0
AVP: l=32 t=Vendor-Specific(26) v=CableLabs(4491)
VSA: l=26 t=CableLabs-QoS-Descriptor(32):
QoS Status: 0x00000005
.............. ........... ......... ....01 = Status Indication: Resource Reserved but not Activated (1)
.............. ........... ......... ....1.. = Service Flow Scheduling Type: 1
.............. ........... ......... ........ 0... = Grant Interval: 0
.............. ........... ......... ........ 0... = Tolerated Grant Jitter: 0
.............. ........... ......... ........ 0... = Grants Per Interval: 0
.............. ........... ......... ........ 0... = Unsolicited Grant Size: 0
.............. ........... ......... ........ 0... = Traffic Priority: 0
.............. ........... ......... ........ 0... = Maximum Sustained Rate: 0
.............. ........... ......... ........ 0... = Maximum Traffic Burst: 0
.............. ........... ......... ........ 0... = Minimum Reserved Traffic Rate: 0
.............. ........... ......... ........ 0... = Minimum Packet Size: 0
.............. ........... ......... ........ 0... = Maximum Concatenated Burst: 0
.............. ........... ......... ........ 0... = Status Request/Transmission Policy: 0
.............. ........... ......... ........ 0... = Nominal Polling Interval: 0
.............. ........... ......... ........ 0... = Tolerated Poll Jitter: 0
.............. ........... ......... ........ 0... = Type of Service Overide: 0
.............. ........... ......... ........ 0... = Maximum Downstream Latency:

Service Class Name:
Service Flow Scheduling Type: 1
AVP: l=12 t=Vendor-Specific(26) v=CableLabs(4491)
VSA: l=6 t=CableLabs-MTA-UDP-Portnum(26): 0
CableLabs-MTA-UDP-Portnum: 0
AVP: l=12 t=Vendor-Specific(26) v=CableLabs(4491)
VSA: l=6 t=CableLabs-SF-ID(30): 0
CableLabs-SF-ID: 0
AVP: l=10 t=Vendor-Specific(26) v=CableLabs(4491)
VSA: l=4 t=CableLabs-Flow-Direction(50): Downstream(2)
CableLabs-Flow-Direction: Downstream (2)
AVP: l=84 t=Vendor-Specific(26) v=CableLabs(4491)
VSA: l=78 t=CableLabs-Event-Message(1):
Event Message Version ID: 4
BCID
Timestamp: 1212445021
Element ID: 0
Time Zone: DST: 1, Offset: +000000
Event Counter: 1
Event Message Type: Call_Answer (15)
Element Type: CMS (1)
Element ID: 0
Time Zone: DST: 1, Offset: +000000
Sequence Number: 4
Event Time: 20080602221701.000
Status: 0x00000008
Example for Event Messages from Cisco Unified Border Element (SP Edition) to RADIUS Billing Server

... .... .... .... .... .... .... 00 = Status: No Error
(0x00000000)
... .... .... .... .... .... .... 0.. = Event Origin: Trusted
Element (0x00000000)
... .... .... .... .... .... .... 1... = Event Message Proxied:
Proxied (0x00000001)
  Priority: 128
  Attribute Count: 2
  Event Object: 0
AVP: l=28 t=Vendor-Specific(26) v=CableLabs(4491)
  VSA: l=22 t=CableLabs-Charge-Number(16): 123
CableLabs-Charge-Number: 123
AVP: l=32 t=Vendor-Specific(26) v=CableLabs(4491)
  VSA: l=26 t=CableLabs-Related-Call-Billing-Correlation-ID(13):
    Timestamp: 1212445021
    Element ID: 0
    Time Zone: DST: 1, Offset: +000000
    Event Counter: 2
AVP: l=84 t=Vendor-Specific(26) v=CableLabs(4491)
  VSA: l=78 t=CableLabs-Event-Message(1):
    Event Message Version ID: 4
    BCID
      Timestamp: 1212445021
      Element ID: 0
      Time Zone: DST: 1, Offset: +000000
      Event Counter: 2
    Event Message Type: Call_Answer (15)
    Element Type: CMS (1)
    Element ID: 0
    Time Zone: DST: 1, Offset: +000000
    Sequence Number: 5
    Status: 0x00000008
    (0x00000000)
    ... .... .... .... .... .... .... 00 = Status: No Error
    (0x00000000)
    ... .... .... .... .... .... .... 0.. = Event Origin: Trusted
    Element (0x00000000)
    ... .... .... .... .... .... .... 1... = Event Message Proxied:
Proxied (0x00000001)
  Priority: 128
  Attribute Count: 2
  Event Object: 0
AVP: l=28 t=Vendor-Specific(26) v=CableLabs(4491)
  VSA: l=22 t=CableLabs-Charge-Number(16): service
CableLabs-Charge-Number: service
AVP: l=32 t=Vendor-Specific(26) v=CableLabs(4491)
  VSA: l=26 t=CableLabs-Related-Call-Billing-Correlation-ID(13):
    Timestamp: 1212445021
    Element ID: 0
    Time Zone: DST: 1, Offset: +000000
    Event Counter: 1
AVP: l=84 t=Vendor-Specific(26) v=CableLabs(4491)
  VSA: l=78 t=CableLabs-Event-Message(1):
    Event Message Version ID: 4
    BCID
      Timestamp: 1212445021
      Element ID: 0
      Time Zone: DST: 1, Offset: +000000
      Event Counter: 1
    Event Message Type: QoS_Commit (19)
    Element Type: CMS (1)
    Element ID: 0
    Time Zone: DST: 1, Offset: +000000
    Sequence Number: 6
Example for Event Messages from Cisco Unified Border Element (SP Edition) to RADIUS Billing Server

Event Time: 20080602221701.000
Status: 0x00000008
.... .... .... .... .... .... .... ..00 = Status: No Error
(0x00000000)
.... .... .... .... .... .... .... .0.. = Event Origin: Trusted Element (0x00000000)
.... .... .... .... .... .... .... 1... = Event Message Proxied: Proxied (0x00000001)
Priority: 128
Attribute Count: 3
Event Object: 0
AVP: l=12 t=Vendor-Specific(26) v=CableLabs(4491)
VSA: l=6 t=CableLabs-MTA-UDP-Portnum(26): 0
CableLabs-MTA-UDP-Portnum: 0
AVP: l=12 t=Vendor-Specific(26) v=CableLabs(4491)
VSA: l=6 t=CableLabs-SF-ID(30): 0
CableLabs-SF-ID: 0
AVP: l=10 t=Vendor-Specific(26) v=CableLabs(4491)
VSA: l=4 t=CableLabs-Flow-Direction(50): Downstream(2)
CableLabs-Flow-Direction: Downstream (2)

Radius Protocol
Code: Accounting-Response (5)
Packet identifier: 0x0 (0)
Example for Event Messages from Cisco Unified Border Element (SP Edition) to RADIUS Billing Server

Length: 20
Authenticator: EB0BD7E187D3301CB7D733A9761F9DE0
[This is a response to a request in frame 1]
[Time from request: 0.041131000 seconds]

No.    Time    Source                Destination           Protocol Info
5 29.324537 77.111.1.51           200.200.1.2           RADIUS
Accounting-Request (4) (id=0, l=1162)

Radius Protocol
Code: Accounting-Request (4)
Packet identifier: 0x0 (0)
Length: 1162
Authenticator: 78D7DE7E0A162046A7936593F80048DS
[The response to this request is in frame 6]
Attribute Value Pairs
AVP: l=6 t=NAS-IP-Address (4): 77.111.1.51
  NAS-IP-Address: 77.111.1.51 (77.111.1.51)
AVP: l=6 t=Acct-Status-Type (40): Interim-Update (3)
  Acct-Status-Type: Interim-Update (3)
AVP: l=26 t=Acct-Session-Id (44): HDq]       01+000000\000\000\000\000\001
  Acct-Session-Id: HDq]       01+000000
AVP: l=84 t=Vendor-Specific (26) v=CableLabs (4491)
  VSA: l=78 t=CableLabs-Event-Message (1):
    Event Message Version ID: 4
    BCID
    Timestamp: 1212445021
    Element ID: 0
    Time Zone: DST: 1, Offset: +000000
    Event Counter: 1
    Event Message Type: Unknown (22)
    Element Type: CMS (1)
    Element ID: 0
    Time Zone: DST: 1, Offset: +000000
    Sequence Number: 8
    Event Time: 20080602221731.000
    Status: 0x00000008
      .... .... .... .... .... .... .... ..00 = Status: No Error
      (0x00000000)
      .... .... .... .... .... .... .... .0.. = Event Origin: Trusted
      Element (0x00000000)
      .... .... .... .... .... .... .... 1... = Event Message Proxied:
      Proxied (0x00000001)
    Priority: 128
    Attribute Count: 1
    Event Object: 0
    AVP: l=134 t=Vendor-Specific (26) v=CableLabs (4491)
      VSA: l=128 t=CableLabs-RTCP-Data (93):
        PS=0, OS=0, PR=0, OR=0, PD=0, OD=0, PL=0, JI=0, LA=0, PC/RPS=0, PC/ROS=0, PC/RPR=0, PC/RPL=0, PC/RJ=0, PC/RP=0, PC/RPS=0, PC/ROS=0, PC/RPR=0, PC/RPL=0, PC/RJ=0
        CableLabs-RTCP-Data: PS=0, OS=0, PR=0, OR=0, PD=0, OD=0, PL=0, JI=0, LA=0, PC/RPS=0, PC/ROS=0, PC/RPR=0, PC/RPL=0, PC/RJ=0
      AVP: l=84 t=Vendor-Specific (26) v=CableLabs (4491)
      VSA: l=78 t=CableLabs-Event-Message (1):
        Event Message Version ID: 4
        BCID
        Timestamp: 1212445021
        Element ID: 0
        Time Zone: DST: 1, Offset: +000000
        Event Counter: 2
Chapter 37      Billing Support

Example for Event Messages from Cisco Unified Border Element (SP Edition) to RADIUS Billing Server

Event Message Type: Unknown (22)
Element Type: CMS (1)
Element ID: 0
Time Zone: DST: 1, Offset: +000000
Sequence Number: 9
Event Time: 20080602221731.000
Status: 0x00000008

.... ..... ...... ...... ........ = Status: No Error
(0x00000000)
.... ..... ...... ...... ...... = Event Origin: Trusted
Element (0x00000000)
.... ..... ...... ...... ...... = Event Message Proxied:
Proxied (0x00000001)
Priority: 128
Attribute Count: 1
Event Object: 0
AVP: l=134  t=Vendor-Specific(26) v=CableLabs(4491)
VSA: l=128 t=CableLabs-RTCP-Data(93): PS=0, OS=0, PR=0, OR=0, PD=0, OD=0,
PL=0, JI=0, LA=0, PC/RPS=0, PC/ROs=0, PC/RPR=0, PC/RPL=0,
PC/RJI=0

CableLabs-RTCP-Data: PS=0, OS=0, PR=0, OR=0, PD=0, OD=0, PL=0, JI=0, LA=0,
PC/RPS=0, PC/ROs=0, PC/RPR=0, PC/RPL=0, PC/RJI=0
AVP: l=84  t=Vendor-Specific(26) v=CableLabs(4491)
VSA: l=78 t=CableLabs-Event-Message(1):
Event Message Version ID: 4
BCID
Timestamp: 1212445021
Element ID: 0
Time Zone: DST: 1, Offset: +000000
Event Counter: 1
Event Message Type: QoS_Release (8)
Element Type: CMS (1)
Element ID: 0
Time Zone: DST: 1, Offset: +000000
Sequence Number: 10
Event Time: 20080602221731.000
Status: 0x00000008

.... ..... ...... ...... ...... ........ = Status: No Error
(0x00000000)
.... ..... ...... ...... ...... = Event Origin: Trusted
Element (0x00000000)
.... ..... ...... ...... ...... = Event Message Proxied:
Proxied (0x00000001)
Priority: 128
Attribute Count: 1
Event Object: 0
AVP: l=12  t=Vendor-Specific(26) v=CableLabs(4491)
VSA: l=6 t=CableLabs-SF-ID(30): 0
CableLabs-SF-ID: 0
AVP: l=10  t=Vendor-Specific(26) v=CableLabs(4491)
VSA: l=4 t=CableLabs-Flow-Direction(50): Upstream(1)
CableLabs-Flow-Direction: Upstream (1)
AVP: l=84  t=Vendor-Specific(26) v=CableLabs(4491)
VSA: l=78 t=CableLabs-Event-Message(1):
Event Message Version ID: 4
BCID
Timestamp: 1212445021
Element ID: 0
Time Zone: DST: 1, Offset: +000000
Event Counter: 2
Event Message Type: QoS_Release (8)
Element Type: CMS (1)
Element ID: 0
Example for Event Messages from Cisco Unified Border Element (SP Edition) to RADIUS Billing Server

Time Zone: DST: 1, Offset: +000000
Sequence Number: 11
Event Time: 20080602221731.000
Status: 0x00000008

    .... .... .... .... .... .... .... ..00 = Status: No Error (0x00000000)

    .... .... .... .... .... .... .... .0.. = Event Origin: Trusted Element (0x00000000)

    .... .... .... .... .... .... .... 1... = Event Message Proxied: Proxied (0x00000001)

        Priority: 128
        Attribute Count: 2
        Event Object: 0
        AVP: l=12 t=Vendor-Specific(26) v=CableLabs(4491)
             VSA: l=6 t=CableLabs-SF-ID(30): 0
             CableLabs-SF-ID: 0
        AVP: l=10 t=Vendor-Specific(26) v=CableLabs(4491)
             VSA: l=4 t=CableLabs-Flow-Direction(50): Downstream(2)
             CableLabs-Flow-Direction: Downstream(2)
        AVP: l=84 t=Vendor-Specific(26) v=CableLabs(4491)
             VSA: l=78 t=CableLabs-Event-Message(1):
                 Event Message Version ID: 4
                 BCID
                 Timestamp: 1212445021
                 Element ID: 0
                 Event Time: 20080602221731.000
                 Status: 0x00000008

        .... .... .... .... .... .... .... ..00 = Status: No Error (0x00000000)

        .... .... .... .... .... .... .... .0.. = Event Origin: Trusted Element (0x00000000)

        .... .... .... .... .... .... .... 1... = Event Message Proxied: Proxied (0x00000001)

        Priority: 128
        Attribute Count: 1
        Event Object: 0
        AVP: l=14 t=Vendor-Specific(26) v=CableLabs(4491)
             VSA: l=8 t=CableLabs-Call-Termination-Cause(11):
                 Source Document: BAF (0x0001)
        AVP: l=84 t=Vendor-Specific(26) v=CableLabs(4491)
             VSA: l=78 t=CableLabs-Event-Message(1):
                 Event Message Version ID: 4
                 BCID
                 Timestamp: 1212445021
                 Element ID: 0
                 Event Time: 20080602221731.000
                 Status: 0x00000008

        .... .... .... .... .... .... .... ..00 = Status: No Error (0x00000000)
Example for Event Messages from Cisco Unified Border Element (SP Edition) to RADIUS Billing Server

```
.... .... .... .... .... .... .... 0.. = Event Origin: Trusted
Element (0x00000000)
.... .... .... .... .... .... .... 1... = Event Message Proxied:
Proxied (0x00000001)
  Priority: 128
  Attribute Count: 2
  Event Object: 0
  AVP: l=32  t=Vendor-Specific(26) v=CableLabs(4491)
  VSA: l=26  t=CableLabs-Related-Call-Billing-Correlation-ID(13):
    Timestamp: 1212445021
    Element ID: 0
    Time Zone: DST: 1, Offset: +000000
    Event Counter: 2
  AVP: l=14  t=Vendor-Specific(26) v=CableLabs(4491)
  VSA: l=8  t=CableLabs-Call-Termination-Cause(11):
    Source Document: BAF (0x0001)
    Event Object: 16
    AVP: l=84  t=Vendor-Specific(26) v=CableLabs(4491)
  VSA: l=78  t=CableLabs-Event-Message(1):
    Event Message Version ID: 4
    BCID
      Timestamp: 1212445021
      Element ID: 0
      Time Zone: DST: 1, Offset: +000000
      Event Counter: 2
      Event Message Type: Call_Disconnect (16)
      Element Type: CMS (1)
      Event Time: 20080602221731.000
      Status: 0x00000008
        .... .... .... .... .... .... .... ..00 = Status: No Error
(0x00000000)
.... .... .... .... .... .... .... 0.. = Event Origin: Trusted
Element (0x00000000)
.... .... .... .... .... .... .... 1... = Event Message Proxied:
Proxied (0x00000001)
  Priority: 128
  Attribute Count: 1
  Event Object: 0
  AVP: l=14  t=Vendor-Specific(26) v=CableLabs(4491)
  VSA: l=8  t=CableLabs-Call-Termination-Cause(11):
    Source Document: BAF (0x0001)
    Event Object: 16
    AVP: l=84  t=Vendor-Specific(26) v=CableLabs(4491)
  VSA: l=78  t=CableLabs-Event-Message(1):
    Event Message Version ID: 4
    BCID
      Timestamp: 1212445021
      Element ID: 0
      Time Zone: DST: 1, Offset: +000000
      Event Counter: 2
      Event Message Type: Signaling_Stop (2)
      Element Type: CMS (1)
      Event Time: 20080602221731.000
      Status: 0x00000008
        .... .... .... .... .... .... .... ..00 = Status: No Error
(0x00000000)
```
Example 2

The following example shows the requests from the SBC to RADIUS server where SBC is configured to include the endpoint adjacency name in billing records:

Mon May 24 10:43:02 2010
NAS-IP-Address = 172.18.53.179
Acct-Status-Type = Interim-Update
Acct-Session-Id = "K\372Ds   326311+030000\000\000\000\t"
CableLabs-Event-Message =
0x00044bfa447320202033323633112b3033330303000000090001000100012020203333233331312b3033330303300000004323031303035323431323834332e3430370000000880000600
Version_ID = 4
Timestamp = 1274692723
Element_ID = 32631
Time_Zone = 1+030000
Event_Counter = 9
Event_Message_Type = Signaling-Start
Element_Type = 1
Element_ID = 32631
Time_Zone = 1+030000
Sequence_Number = 64
Event_Time = 20100524121843.407
Status = 8
Priority = 128
Attribute_Count = 6
Event_Object = 0
CableLabs-Direction-indicator = 0x0001
Originating
CableLabs-WTA-Endpoint-Name = "SIPPB"
CableLabs-Calling-Party-Number = "sipp"
CableLabs-Called-Party-Number = "service"
CableLabs-Routing-Number = "service"
CableLabs-Attr-87 = 0x0003
Example for Event Messages from Cisco Unified Border Element (SP Edition) to RADIUS Billing Server

Billing type -
flat rate
CableLabs-Event-Message =
0x00044bfa44732020203332363331312b303330300000000a000100012020332363331312b30333030
301000000004132303130303532343132313834332e34303700000008800000600
Version_ID = 4
Timestamp = 1274692723
Element_ID = 32631
Time_Zone = 1+030000
Event.Counter = 10

**Event_Message_Type = Signaling-Start**

Element_Type = 1
Element_ID = 32631
Time_Zone = 1+030000
Sequence_Number = 65
Event_Time = 20100524121843.407
Status = 8
Priority = 128
Attribute_Count = 6
Event_Object = 0

CableLabs-Direction-indicator = 0x0002
Terminating

**CableLabs-MTA-Endpoint-Name = "MTA Endpoint"**

CableLabs-Calling-Party-Number = "sipp"
CableLabs-Called-Party-Number = "service"
CableLabs-Routing-Number = "service"
CableLabs-Attr-87 = 0x0003
Billing type -
flat rate
Acct.Unique-Session-Id = "3479bc93d50898b5"
Timestamp = 1274712182
Request-Authenticator = Verified

Mon May 24 10:43:02 2010
NAS-IP-Address = 172.18.53.179
Acct-Status-Type = Interim-Update
Acct-Session-Id = "K\372Ds 326311+030000\000\000\000\t"
CableLabs-Event-Message =
0x00044bfa44732020203332363331312b303330303000000090007000120203332363331312b30333030
301000000004132303130303532343132313834332e34303700000008800000400
Version_ID = 4
Timestamp = 1274692723
Element_ID = 32631
Time_Zone = 1+030000
Event.Counter = 9

**Event_Message_Type = QoS-Reserve**

Element_Type = 1
Element_ID = 32631
Time_Zone = 1+030000
Sequence_Number = 66
Event_Time = 20100524121843.407
Status = 8
Priority = 128
Attribute_Count = 4
Event_Object = 0

CableLabs-QoS-Descriptor = 0x00000005202020202020202020202020202020202020202020202020202020
Status_Bitmask = 5
Service_Class_Name =
QoS_Parameter_Array = 1
resource reserved but not committed
CableLabs-MTA-UDP-Portnum = 0
CableLabs-SF-ID = 0
CableLabs-Flow-Direction = 0x0001
Upstream
Example for Event Messages from Cisco Unified Border Element (SP Edition) to RADIUS Billing Server

CableLabs-Event-Message =
0x00004bfa473202020332363331312b303330303000000000a0007000120203332363331312b30333030300000000000000000088000000400

Version_ID = 4
Timestamp = 1274692723
Element_ID = 32631
Event_Counter = 10
Event_Message_Type = QoS-Reserve
Element_Type = 1
Element_ID = 32631
Time_Zone = 1+030000
Sequence_Number = 67
Event_Time = 20100524121843.407
Status = 8
Priority = 128
Attribute_Count = 4
Event_Object = 0

CableLabs-QoS-Descriptor = 0x000000052020202020202020202020202020202000000001
Status_Bitmap = 5
Service_Class_Name =
QoS_Parameter_Array = 1
resource reserved but not committed

CableLabs-MTA-UDP-Portnum = 0
CableLabs-SF-ID = 0
CableLabs-Flow-Direction = 0x0002
Downstream
Acct-Unique-Session-Id = "3479bc93d50898b5"
Timestamp = 1274712182
Request-Authenticator = Verified

Mon May 24 10:43:02 2010
NAS-IP-Address = 172.18.53.179
Acct-Status-Type = Interim-Update
Acct-Session-Id = "K\372Ds   326311+030000\000\000\000\000\t"

CableLabs-Event-Message =
0x00004bfa473202020332363331312b303330303000000000a0007000120203332363331312b30333030300000000000000000088000000300

Version_ID = 4
Timestamp = 1274692723
Element_ID = 32631
Time_Zone = 1+030000
Event_Counter = 9
Event_Message_Type = Call-Answer
Element_Type = 1
Element_ID = 32631
Time_Zone = 1+030000
Sequence_Number = 68
Event_Time = 20100524121843.407
Status = 8
Priority = 128
Attribute_Count = 2
Event_Object = 0

CableLabs-Charge-Number = "sip:
CableLabs-Related-Call-Billing-Crl-ID =
0x4bfa473202020332363331312b3033303030000000000a0007000120203332363331312b303330303000000000000000000880000000a

Timestamp = 1274692723
Element_ID = 32631
Time_Zone = 1+030000
Event_Counter = 10

CableLabs-Event-Message =
0x00004bfa473202020332363331312b303330303000000000a0007000120203332363331312b303330303000000000000000000880000000a

Version_ID = 4
Example for Event Messages from Cisco Unified Border Element (SP Edition) to RADIUS Billing Server

Chapter 37      Billing Support

Example for Event Messages from Cisco Unified Border Element (SP Edition) to RADIUS Billing Server

Timestamp = 1274692723
Element_ID = 32631
Time_Zone = 1+030000
Event_Counter = 10

Event_Message_Type = Call-Answer
Element_Type = 1
Element_ID = 32631
Time_Zone = 1+030000
Sequence_Number = 69
Event_Time = 20100524121843.407
Status = 8
Priority = 128
Attribute_Count = 3
Event_Object = 0
CableLabs-Charge-Number = "service"
CableLabs-Related-Call-Billing-Crl-ID = 0xdbaf4732020333236331312b3033303030000000009

Timestamp = 1274692723
Element_ID = 32631
Time_Zone = 1+030000
Event_Counter = 9

CableLabs-MTA-Endpoint-Name = "SIPPA"
Acct-Unique-Session-Id = "3479bc93d50898b5"
Timestamp = 1274712182
Request-Authenticator = Verified

Mon May 24 10:43:02 2010
NAS-IP-Address = 172.18.53.179
Acct-Status-Type = Interim-Update
Acct-Session-Id = "K\372Ds   326311+030000\000\000\000	"
CableLabs-Event-Message = 0x00044bfa44732020333236331312b30333030300000000090013000120020333236331312b3033303030100000004632303130303532343132313834332e343037000000000880000300

Version_ID = 4
Timestamp = 1274692723
Element_ID = 32631
Time_Zone = 1+030000
Event_Counter = 9
Event_Message_Type = QoS-Commit
Element_Type = 1
Element_ID = 32631
Time_Zone = 1+030000
Sequence_Number = 70
Event_Time = 20100524121843.407
Status = 8
Priority = 128
Attribute_Count = 3
Event_Object = 0
CableLabs-MTA-UDP-Portnum = 0
CableLabs-SF-ID = 0
CableLabs-Flow-Direction = 0x0001
Upstream
CableLabs-Event-Message = 0x00044bfa44732020333236331312b30333030300000000090013000120020333236331312b3033303030100000004732303130303532343132313834332e343037000000000880000300

Version_ID = 4
Timestamp = 1274692723
Element_ID = 32631
Time_Zone = 1+030000
Event_Counter = 10
Event_Message_Type = QoS-Commit
Element_Type = 1
Element_ID = 32631
Time_Zone = 1+030000
Example for Event Messages from Cisco Unified Border Element (SP Edition) to RADIUS Billing Server

Sequence_Number = 71
Event_Time = 20100524121843.407
Status = 8
Priority = 128
Attribute_Count = 3
Event_Object = 0
CableLabs-MTA-UDP-Portnum = 0
CableLabs-SF-ID = 0
CableLabs-Flow-Direction = 0x0002
  Downstream
Acct-Unique-Session-Id = "3479bc93d50898b5"
Timestamp = 1274712182
Request-Authenticator = Verified

Mon May 24 10:43:02 2010
NAS-IP-Address = 172.18.53.179
Acct-Status-Type = Interim-Update
Acct-Session-Id = "K\372Ds   326311+030000\000\000\000\t"
CableLabs-Event-Message =
  0x00044bfa447320202033323633331312b30333030300000000900160001202033323633331312b30333030
  3030000000043832303130303533243133231384332e343037000000880000100
  Version_ID = 4
  Timestamp = 1274692723
  Element_ID = 32631
  Time_Zone = 1+030000
  Event_Counter = 9
  Event_Message_Type = Media-Statistics
  Element_Type = 1
  Element_ID = 32631
  Time_Zone = 1+030000
  Sequence_Number = 72
  Event_Time = 20100524121843.407
  Status = 8
  Priority = 128
  Attribute_Count = 1
  Event_Object = 0
  CableLabs-Attr-93 =
    0x50533d302c204f533d302c204f533d302c20505233d302c204f443d302c20504c3d302c204a49
    3d302c204c413d302c2050432f5250533d302c2050432f524f533d302c2050432f5250533d302c2050432f5250
    4c3d302c2050432f524a493d30
  RTP Data:
    PS=0, OS=0, PR=0, OR=0, PD=0, OD=0, PL=0, JI=0, LA=0, PC/RPS=0, PC/ROS=0,
    PC/RPR=0, PC/RPL=0, PC/RJ=0
  CableLabs-Event-Message =
    0x00044bfa447320202033323633331312b30333030300000000a0160001202033323633331312b30333030
    3030000000049323031303533243133231384332e343037000000880000100
  Version_ID = 4
  Timestamp = 1274692723
  Element_ID = 32631
  Time_Zone = 1+030000
  Event_Counter = 10
  Event_Message_Type = Media-Statistics
  Element_Type = 1
  Element_ID = 32631
  Time_Zone = 1+030000
  Sequence_Number = 73
  Event_Time = 20100524121843.407
  Status = 8
  Priority = 128
  Attribute_Count = 1
  Event_Object = 0
Example for Event Messages from Cisco Unified Border Element (SP Edition) to RADIUS Billing Server

CableLabs-Attr-93 = 0x50533d302c204f533d302c2050523d302c20505443d302c20504c3d302c204a493d302c204c413d302c2050432f5250533d302c2050432f524f533d302c2050432f5250533d302c2050432f52504c3d302c2050432f5250493d30
  RTCP Data:
  PS=0, OS=0, PR=0, OR=0, PD=0, OD=0, PL=0, JI=0, LA=0, PC/RPS=0, PC/ROS=0,
  PC/RPR=0, PC/RPL=0, PC/RJI=0
  Acct-Unique-Session-Id = "3479bc93d50898b5"
  Timestamp = 1274712182
  Request-Authenticator = Verified

Mon May 24 10:43:02 2010
NAS-IP-Address = 172.18.53.179
Acct-Status-Type = Interim-Update
Acct-Session-Id = "K\372Ds 326311+030000\000\000\000\t"
CableLabs-Event-Message =
  Version_ID = 4
  Timestamp = 1274692723
  Element_ID = 32631
  Time_Zone = 1+030000
  Event_Counter = 9
  Event_Message_Type = QoS-Release
  Element_Type = 1
  Time_Zone = 1+030000
  Sequence_Number = 74
  Event_Time = 20100524121843.407
  Status = 8
  Priority = 128
  Attribute_Count = 2
  Event_Object = 0
  CableLabs-SF-ID = 0
  CableLabs-Flow-Direction = 0x0001
  Upstream

CableLabs-Event-Message =
  Version_ID = 4
  Timestamp = 1274692723
  Element_ID = 32631
  Time_Zone = 1+030000
  Event_Counter = 10
  Event_Message_Type = QoS-Release
  Element_Type = 1
  Time_Zone = 1+030000
  Sequence_Number = 75
  Event_Time = 20100524121843.407
  Status = 8
  Priority = 128
  Attribute_Count = 2
  Event_Object = 0
  CableLabs-SF-ID = 0
  CableLabs-Flow-Direction = 0x0002
  Downstream

Mon May 24 10:43:02 2010
NAS-IP-Address = 172.18.53.179
Acct-Status-Type = Interim-Update
Example for Event Messages from Cisco Unified Border Element (SP Edition) to RADIUS Billing Server

Acct-Session-Id = "K\372Ds 32631\030000\000\000\000\0t"
CableLabs-Event-Message =
0x00044bfa447320203332363331312b30333303030300000000900100001202033323633331312b303333030300000000004c32303130303532343132313834332e3430370000000880000100
Version_ID = 4
Timestamp = 1274692723
Element_ID = 32631
Time_Zone = 1+030000
Event_Counter = 9

**Event_Message_Type = Call-Disconnect**

Element_Type = 1
Element_ID = 32631
Time_Zone = 1+030000
Sequence_Number = 76
Event_Time = 20100524121843.407
Status = 8
Priority = 128
Attribute_Count = 1
Event_Object = 0

CableLabs-Call-Termination-Cause = 0x000100000010
Cause: Normal call clearing

CableLabs-Event-Message =
0x00044bfa447320203332363331312b303333030303000000009000200001202033323633331312b3033330303000000004d32303130303532343132313834332e3430370000000880000300
Version_ID = 4
Timestamp = 1274692723
Element_ID = 32631
Time_Zone = 1+030000
Event_Counter = 9

**Event_Message_Type = Signaling-Stop**

Element_Type = 1
Element_ID = 32631
Time_Zone = 1+030000
Sequence_Number = 77
Event_Time = 20100524121843.407
Status = 8
Priority = 128
Attribute_Count = 3
Event_Object = 0

CableLabs-Related-Call-Billing-Crl-ID =
0x4bfa447320203332363331312b303333030303000000009000200001202033323633331312b3033330303000000004e32303130303532343132313834332e3430370000000880000100
Timestamp = 1274692723
Element_ID = 32631
Time_Zone = 1+030000
Event_Counter = 10

CableLabs-Call-Termination-Cause = 0x000100000010
Cause: Normal call clearing

CableLabs-MTA-Endpoint-Name = "SIPPB"

CableLabs-Event-Message =
0x00044bfa447320203332363331312b30333303030300000000900100001202033323633331312b303333030300000000004d32303130303532343132313834332e3430370000000880000100
Version_ID = 4
Timestamp = 1274692723
Element_ID = 32631
Time_Zone = 1+030000
Event_Counter = 10

**Event_Message_Type = Call-Disconnect**

Element_Type = 1
Element_ID = 32631
Time_Zone = 1+030000
Sequence_Number = 78
Event_Time = 20100524121843.407
Status = 8
Priority = 128
Example 3

The following example shows requests from the SBC to RADIUS server where SBC is configured to include endpoint addressing information in billing records:

Tue May 11 13:26:00 2010
NAS-IP-Address = 172.18.53.179
Acct-Status-Type = Interim-Update
Acct-Session-Id = "K\351GA 326311+03000\000\000\001"
CableLabs-Event-Message =
0x4bfa473202020333236333132b3033303030000000000000a00020001202020333236333132b30333030
30100000004f32303130303532343132313834332e3430370000000088000000
Version_ID = 4
Timestamp = 1273579329
Element_ID = 32631
Time_Zone = 1+030000
Event_Counter = 0

Event_Message_Type = Signaling-Start
Element_Type = 1
Element_ID = 32631
Sequence_Number = 0
Event_Time = 20100511150208.964
Status = 8
Priority = 128
Attribute_Count = 6

Example for Event Messages from Cisco Unified Border Element (SP Edition) to RADIUS Billing Server
Example for Event Messages from Cisco Unified Border Element (SP Edition) to RADIUS Billing Server

<table>
<thead>
<tr>
<th>Event_Object = 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>CableLabs-Direction-indicator = 0x0001</td>
</tr>
</tbody>
</table>

**Originating**

<table>
<thead>
<tr>
<th>CableLabs-MTA-Endpoint-Name = &quot;2.0.0.36,5078,UDP,SIPPB&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>CableLabs-Calling-Party-Number = &quot;sipp&quot;</td>
</tr>
<tr>
<td>CableLabs-Called-Party-Number = &quot;service&quot;</td>
</tr>
<tr>
<td>CableLabs-Routing-Number = &quot;service&quot;</td>
</tr>
<tr>
<td>CableLabs-Attr-87 = 0x0003</td>
</tr>
</tbody>
</table>

Billing type: flat rate

CableLabs-Event-Message =

```
0x00044be94741202020333236333132b303333030300000002000100012020333236333132b303330303000000000000008800000000
```

**Version_ID = 4**
**Timestamp = 1273579329**
**Element_ID = 32631**
**Time_Zone = 1+030000**
**Event_Counter = 2**

<table>
<thead>
<tr>
<th>Event_Message_Type = Signaling-Start</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element_Type = 1</td>
</tr>
<tr>
<td>Element_ID = 32631</td>
</tr>
<tr>
<td>Time_Zone = 1+030000</td>
</tr>
<tr>
<td>Sequence_Number = 1</td>
</tr>
<tr>
<td>Event_Time = 20100511150208.964</td>
</tr>
<tr>
<td>Status = 8</td>
</tr>
<tr>
<td>Priority = 128</td>
</tr>
<tr>
<td>Attribute_Count = 6</td>
</tr>
</tbody>
</table>

**Event_Object = 0**

**Terminating**

<table>
<thead>
<tr>
<th>CableLabs-Direction-indicator = 0x0002</th>
</tr>
</thead>
<tbody>
<tr>
<td>CableLabs-MTA-Endpoint-Name = &quot;MTA Endpoint&quot;</td>
</tr>
<tr>
<td>CableLabs-Calling-Party-Number = &quot;sipp&quot;</td>
</tr>
<tr>
<td>CableLabs-Called-Party-Number = &quot;service&quot;</td>
</tr>
<tr>
<td>CableLabs-Routing-Number = &quot;service&quot;</td>
</tr>
<tr>
<td>CableLabs-Attr-87 = 0x0003</td>
</tr>
</tbody>
</table>

Billing type: flat rate

Acct-Unique-Session-Id = "95a26a97e3e0c3c"
Timestamp = 1273598760
Request-Authenticator = Verified
Example for Event Messages from Cisco Unified Border Element (SP Edition) to RADIUS Billing Server

CableLabs-QoS-Descriptor = 0x00000005202020202020202020202020200000001
  Status_Bitmask = 5
  Service_Class_Name =
  QoS_Parameter_Array = 1
    resource reserved but not committed
CableLabs-MTA-UDP-Portnum = 0
CableLabs-SF-ID = 0
CableLabs-Flow-Direction = 0x0001
  Upstream
CableLabs-Event-Message =
  0x00044be947412020203332363331312b303330303030000000200070001202020332363331312b30333030
  303000000003323031303035313135303230392e30373330000000380000400
    Version_ID = 4
    Timestamp = 1273579329
    Element_ID = 32631
    Time_Zone = 1+030000
    Event.Counter = 2
    Event_Message_Type = QoS-Reserve
    Element_Type = 1
    Element_ID = 32631
    Time_Zone = 1+030000
    Sequence_Number = 3
    Event_Time = 20100511150209.073
    Status = 8
    Priority = 128
    Attribute_Count = 4
    Event_Object = 0
CableLabs-QoS-Descriptor = 0x00000005202020202020202020202020200000001
  Status_Bitmask = 5
  Service_Class_Name =
  QoS_Parameter_Array = 1
    resource reserved but not committed
CableLabs-MTA-UDP-Portnum = 0
CableLabs-SF-ID = 0
CableLabs-Flow-Direction = 0x0002
  Downstream
Acct-Unique-Session-Id = "95a26a97e3e08c3c"
  Timestamp = 1273598759
  Request Authenticator = Verified

Tue May 11 13:26:00 2010
NAS-IP-Address = 172.18.53.179
Acct-Status-Type = Interim-Update
Acct-Session-Id = "K\351GA   326311+030000\000\000\000\001"
CableLabs-Event-Message =
  0x00044be947412020203332363331312b303330303030000000200070001202020332363331312b30333030
  303000000003323031303035313135303230392e35373330000000380000300
    Version_ID = 4
    Timestamp = 1273598759
    Element_ID = 32631
    Time_Zone = 1+030000
    Event.Counter = 1
    Event_Message_Type = Call-Answer
    Element_Type = 1
    Element_ID = 32631
    Time_Zone = 1+030000
    Sequence_Number = 4
    Event_Time = 20100511150209.573
    Status = 8
    Priority = 128
    Attribute_Count = 2
    Event_Object = 0
CableLabs-Charge-Number = "sipp"
Example for Event Messages from Cisco Unified Border Element (SP Edition) to RADIUS Billing Server

CableLabs-Related-Call-Billing-Crl-ID = 0x4be947412020203332363331312b303330303000000002
Timestamp = 1273579329
Element_ID = 32631
Time_Zone = 1+030000
Event_Counter = 2

CableLabs-Event-Message = 0x00044be947412020203332363331312b3033303030000000002000f00012020203332363331312b30333030300000000002000f00012020203332363331312b30333030300000000002000f00012020203332363331312b30333030300000000001
Version_ID = 4
Timestamp = 1273579329
Element_ID = 32631
Time_Zone = 1+030000
Event_Counter = 2

Event_Message_Type = Call-Answer
Element_Type = 1
Element_ID = 32631
Time_Zone = 1+030000
Sequence_Number = 5
Event_Time = 2010051150209.573
Status = 8
Priority = 128
Attribute_Count = 3
Event_Object = 0

CableLabs-Charge-Number = "service"
CableLabs-Related-Call-Billing-Crl-ID = 0x4be947412020203332363331312b303330303000000001
Timestamp = 1273579329
Element_ID = 32631
Time_Zone = 1+030000
Event_Counter = 1

CableLabs-MTA-Endpoint-Name = "1.0.0.36,5068,UDP,SIPPA"
Acct-Unique-Session-Id = "95a26a97e3e08c3c"
Timestamp = 1273598760
Request-Authenticator = Verified

Tue May 11 13:26:00 2010
NAS-IP-Address = 172.18.53.179
Acct-Status-Type = Interim-Update
Acct-Session-Id = "K\351GA 326311+030000\000\000\000\001"

CableLabs-Event-Message = 0x00044be947412020203332363331312b30333030300000000002000f00012020203332363331312b30333030300000000002000f00012020203332363331312b30333030300000000002000f00012020203332363331312b30333030300000000001
Version_ID = 4
Timestamp = 1273579329
Element_ID = 32631
Time_Zone = 1+030000
Event_Counter = 1

Event_Message_Type = QoS-Commit
Element_Type = 1
Element_ID = 32631
Time_Zone = 1+030000
Sequence_Number = 6
Event_Time = 2010051150209.573
Status = 8
Priority = 128
Attribute_Count = 3
Event_Object = 0

CableLabs-MTA-UDP-Portnum = 0
CableLabs-SF-ID = 0
CableLabs-Flow-Direction = 0x0001
Upstream
Example for Event Messages from Cisco Unified Border Element (SP Edition) to RADIUS Billing Server

CableLabs-Event-Message = 0x00044be947412020203333312b303333030300000000000020013000120202003332363331312b3033303030100000000007323031303531313133303230392e3537330000008800000300
Version_ID = 4
Timestamp = 1273579329
Element_ID = 32631
Time_Zone = 1+030000
Event_Counter = 2
Event_Message_Type = QoS-Commit
Element_Type = 1
Element_ID = 32631
Time_Zone = 1+030000
Sequence_Number = 7
Event_Time = 20100511150209.573
Status = 8
Priority = 128
Attribute_Count = 3
Event_Object = 0
CableLabs-MTA-UDP-Portnum = 0
CableLabs-SF-ID = 0
CableLabs-Flow-Direction = 0x0002
Downstream
Acct-Unique-Session-Id = "95a26a97e08c3c"
Timestamp = 1273598760
Request-Authenticator = Verified

Tue May 11 13:26:00 2010
NAS-IP-Address = 172.18.53.179
Acct-Status-Type = Interim-Update
Acct-Session-Id = "K\351GA   326311+030000\000\000\001"
CableLabs-Event-Message = 0x00044be947412020203333312b303333030300000000000010016000120202003332363331312b3033303030100000000008323031303531313133303230392e3537330000008800000100
Version_ID = 4
Timestamp = 1273579329
Element_ID = 32631
Time_Zone = 1+030000
Event_Counter = 1
Event_Message_Type = Media-Statistics
Element_Type = 1
Element_ID = 32631
Time_Zone = 1+030000
Sequence_Number = 8
Event_Time = 20100511150209.573
Status = 8
Priority = 128
Attribute_Count = 3
Event_Object = 0
CableLabs-Attr-93 = 0x50533d302c204f533d302c2050523d302c204f523d302c2050443d302c204f443d302c20504c3d302c204a493d302c204c413d302c2050432f5250533d302c2050432f5250533d302c2050432f5250523d302c2050432f5250
4c1d302c2050432f524a493d30 RTCP Data:
PS=0, OS=0, PR=0, OR=0, PD=0, OD=0, PL=0, JJ=0, LA=0, PC/RPS=0, PC/ROS=0, PC/RPR=0, PC/RPL=0, PC/RIJ=0
CableLabs-Event-Message = 0x00044be947412020203333312b303333030300000000000020016000120202003332363331312b3033303030100000000008323031303531313133303230392e3537330000008800000100
Version_ID = 4
Timestamp = 1273579329
Element_ID = 32631
Time_Zone = 1+030000
Event_Counter = 2
Event_Message_Type = Media-Statistics

Example for Event Messages from Cisco Unified Border Element (SP Edition) to RADIUS Billing Server

```
Element_Type = 1
Element_ID = 32631
Time_Zone = 1+030000
Sequence_Number = 9
Event_Time = 20100511150209.573
Status = 8
Priority = 128
Attribute_Count = 1
Event_Object = 0

CableLabs-Attr-93 =
0x50533d302c204f533d302c2050523d302c204f523d302c2050443d302c204f443d302c20504c3d302c204f523d302c2050432f524f533d302c2050432f5250523d302c2050432f5250
RTCP Data:
PS=0, OS=0, PR=0, OR=0, PD=0, OD=0, PL=0, JI=0, LA=0, PC/RPS=0, PC/RPS=0,
PC/RPL=0, PC/RPL=0,
Acct-Unique-Session-Id = "95a26a97e3e08c3c"
Request-Authenticator = Verified
Tue May 11 13:26:00 2010
NAS-IP-Address = 172.18.53.179
Acct-Status-Type = Interim-Update
Acct-Session-Id = "K\351GA 32631+030000\000\000\000\000"  
CableLabs-Event-Message =
0x00044be947412020203332363331312b303333030300000000000000012002023332363331312b30333303
30300000003a323031303533131335303230392e353733000000088000000200
Version_ID = 4
Timestamp = 1273579329
Element_ID = 32631
Time_Zone = 1+030000
Event_Counter = 1
Event_Message_Type = QoS-Release
Element_Type = 1
Element_ID = 32631
Time_Zone = 1+030000
Sequence_Number = 10
Event_Time = 20100511150209.573
Status = 8
Priority = 128
Attribute_Count = 2
Event_Object = 0

CableLabs-SF-ID = 0
CableLabs-Flow-Direction = 0x0001
Upstream
CableLabs-Event-Message =
0x00044be947412020203332363331312b303333030300000000000000012002023332363331312b30333303
30300000003a323031303533131335303230392e353733000000088000000200
Version_ID = 4
Timestamp = 1273579329
Element_ID = 32631
Time_Zone = 1+030000
Event_Counter = 2
Event_Message_Type = QoS-Release
Element_Type = 1
Element_ID = 32631
Time_Zone = 1+030000
Sequence_Number = 11
Event_Time = 20100511150209.573
Status = 8
Priority = 128
Attribute_Count = 2
Event_Object = 0

CableLabs-SF-ID = 0
```
Example for Event Messages from Cisco Unified Border Element (SP Edition) to RADIUS Billing Server

CableLabs-Flow-Direction = 0x0002
Downstream
Acct-Unique-Session-Id = "95a26a97e3e08c3c"
Timestamp = 1273598760
Request-Authenticator = Verified

Tue May 11 13:26:00 2010
NAS-IP-Address = 172.18.53.179
Acct-Status-Type = Interim-Update
Acct-Session-Id = "K\351GA   326311+030000\000\000\000\001"
CableLabs-Event-Message =
0x00044be9474120200332363331312b3033303030300000010010001202020332363331312b30333030
30100000000c32303130303531313135303230392e35373300000008800003100
Version_ID = 4
Timestamp = 1273579329
Element_ID = 32631
Time_Zone = 1+030000
Event_Counter = 1

**Event Message Type = Call-Disconnect**
Element_Type = 1
Element_ID = 32631
Time_Zone = 1+030000
Sequence_Number = 12
Event_Time = 20100511150209.573
Status = 8
Priority = 128
Attribute_Count = 1
Event_Object = 0
CableLabs-Call-Termination-Cause = 0x000100000010
Cause: Normal call clearing

CableLabs-Event-Message =
0x00044be9474120200332363331312b3033303030300000010010001202020332363331312b30333030
30100000000c32303130303531313135303230392e35373300000008800003100
Version_ID = 4
Timestamp = 1273579329
Element_ID = 32631
Time_Zone = 1+030000
Event_Counter = 1

**Event Message Type = Signaling-Stop**
Element_Type = 1
Element_ID = 32631
Time_Zone = 1+030000
Sequence_Number = 13
Event_Time = 20100511150209.573
Status = 8
Priority = 128
Attribute_Count = 3
Event_Object = 0
CableLabs-Related-Call-Billing-Crl-ID =
0x4be94741202000332363331312b3033303030000000000000000000010010001202020332363331312b30333030
30100000000c32303130303531313135303230392e353733000000088000010010
Version_ID = 4
Timestamp = 1273579329
Element_ID = 32631
Time_Zone = 1+030000
Event_Counter = 2
CableLabs-Call-Termination-Cause = 0x000100000010
Cause: Normal call clearing

CableLabs-MTA-Endpoint-Name = "2.0.0.36,5078,UDP,SIPPB"
CableLabs-Event-Message =
0x00044be9474120200332363331312b3033303030000000000000000000010010001202020332363331312b30333030
30100000000c32303130303531313135303230392e353733000000088000010010
Version_ID = 4
Timestamp = 1273579329
Element_ID = 32631
Time_Zone = 1+030000


Security

The PacketCable 1.5 Event Messages Specification mandates that the billing messages are sent using the RADIUS protocol and IPSec for security.

**Note**

In ACE SBC Release 3.0.00, only the RADIUS security mechanism, based on its own Request Authenticator, is supported.
Secure Media and SRTP Passthrough

Cisco Unified Border Element (SP Edition) supports two methods of encrypted data streams—Secure Real-Time Protocol (SRTP) Passthrough and Secure Media. The preferred method is to use SRTP Passthrough because it allows the end points themselves to signal their encryption capabilities.

The Secure Media feature is enabled on the global level for all calls and is disabled by default. When Secure Media is turned on globally, the SBC assumes that all end points are going to use encrypted data streams regardless of the actual end point capabilities.

Starting with Cisco IOS XE Release 2.6, using the Unsignaled Secure Media feature you are able to configure secure media on a granular level for specific calls and adjacencies using Call Admission Control (CAC) table entry commands.

You can configure SRTP Passthrough on a granular basis using CAC policy.

Regardless of the method used to configure the Cisco Unified Border Element (SP Edition) to accept encrypted media packets, Cisco Unified Border Element (SP Edition) reserves additional bandwidth to ensure these packets pass through. Typically, the bandwidth of a media stream is determined by the codecs that the endpoints use. However, the use of the encryption in the media streams increases the packet size. As a rule of thumb, the bandwidth requirements are 10% more than the unencrypted codec. However, this increase is not reflected in the media flow statistics.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).

For a complete description of the commands used in this chapter, refer to the Cisco Unified Border Element (SP Edition) Command Reference: Unified Model at:

For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

Feature History for Secure Media and SRTP Passthrough

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 2.4</td>
<td>These features were introduced on the Cisco ASR 1000 Series Aggregation Services Routers.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 2.6</td>
<td>The Unsignaled Secure Media feature was introduced to allow configuration at a granular level using CAC table entry commands. With the introduction of this feature, the Configuring Secure Media-Global Level feature has been deprecated.</td>
</tr>
</tbody>
</table>
Chapter 38  Secure Media and SRTP Passthrough

This chapter contains the following sections:

- Prerequisites for Secure Media and SRTP Passthrough, page 38-2
- Restrictions for Secure Media, page 38-2
- Information About Secure Media, page 38-3
- Information About SRTP Passthrough, page 38-4
- Information About SRTP to RTP Interworking and SRTP Passthrough, page 38-7
- Configuring Secure Media—Global Level, page 38-12
- Configuring Unsignaled Secure Media at a Granular Level, page 38-13
- Configuring SRTP Passthrough, page 38-18
- Configuring CAC Policies for SRTP to RTP Interworking, page 38-23
- SRTP Support for RTCP Multiplexed with RTP, page 38-28
- SRTP Support for SSRC-Based Multiplexing, page 38-29
- Configuring Global Secure Media Example, page 38-29
- Configuring Unsignaled, Granular-Level Secure Media: Examples, page 38-30
- Configuring SRTP Passthrough Example, page 38-32
- CAC Policies for SRTP to RTP Interworking Configuration: Example, page 38-33

Prerequisites for Secure Media and SRTP Passthrough

The following prerequisites are required to implement both features:

Before implementing the Secure Media and SRTP Passthrough features, Cisco Unified Border Element (SP Edition) must already be configured.

Restrictions for Secure Media

The following is a restriction for Global and Unsignaled Secure Media:

- With this feature enabled, RTCP related statistics displayed in the `show sbc dbe media-flow-stats` command are displayed as unknown.

The following is a restriction for Unsignaled (granular-level) Secure Media:

- Both caller and callee sides of the call need to be configured with the `caller secure-media` and `callee secure-media` commands. If only one leg of the call is configured, then the call will fail.
Information About Secure Media

Typically, an endpoint will indicate that the media traffic is encrypted through the SIP signaling. The encryption keys are either exchanged through Session Description Protocol (SDP) or using the Datagram Transport Layer Security (DTLS) mechanism.

In Cisco IOS XE Release 2.4 and Release 2.5, Cisco Unified Border Element (SP Edition) interworked with endpoints or SIP devices that use encrypted media (DTLS or Secure-RTP [SRTP]), but the endpoints did not indicate this in the SIP signaling. In those earlier releases, the SBC supported a globally enabled Secure Media configuration where all calls on the SBC were treated as consisting of SRTP media. Even though the endpoint may not have signaled for SRTP media, media pinholes were created as if the traffic was SRTP. A global configuration under the SBE submode indicates that the endpoints are using encrypted SRTP media, but they will not be using SIP signaling to communicate and negotiate as such. The consequence of this configuration being applied at a global level is that even for flows that are not encrypted, additional bandwidth is reserved and RTP and RTCP checking and validations are disabled.

When interworking with a SIP device that does not have full support for signaling SRTP media streams, the SBC cannot know in advance that the media will be SRTP because it is not signaled as SRTP. Starting with Cisco IOS XE Release 2.6, the Unsignaled Secure Media feature allows the SBC to successfully interoperate with SIP devices that generate SRTP media but signal this as a regular RTP media stream.

You are able to configure the SBC to know which SIP devices it communicates with require support for unsignaled SRTP. Such SIP devices are assumed to always send SRTP media. Minimally you must granularly configure all devices on a given adjacency to require support for SRTP. In configuring secure media on a granular level, you use Call Admission Control (CAC) table entry commands. We highly recommend you use the granular level configuration because, instead of turning on secure media globally, you can specify the calls and adjacencies where you want to use secure media. Using the granular option of Unsignaled Secure Media, additional bandwidth is allocated and RTCP no check is performed only for those calls that match the CAC match criteria. Unsignaled Secure Media, like the global option, is disabled by default.

In Cisco IOS XE Release 2.6, when you configure the SBC to allow unsignaled SRTP media on a granular level for adjacencies, observe these recommended guidelines:

- If the adjacencies are trusted to allow secure calls—use either the `security trusted-encrypted` or `security trusted-unencrypted` command to configure both adjacencies where caller and callee side are located for SRTP passthrough first. Both sides need to be configured because it is a passthrough. This is the default where SRTP calls are allowed between trusted adjacencies.
- If an adjacency is not trusted, you can still configure granular-level Unsignaled Secure Media on that adjacency by configuring SRTP Passthrough in a CAC configuration on the untrusted adjacency. Use the `srtp support` command to allow an SRTP call on the adjacency where the CAC policy is applied.
- Configure both legs of the call to enable the granular-level Unsignaled Secure Media—use the `caller secure-media` command on the caller side, and the `callee secure-media` command on the callee side.
Information About SRTP Passthrough

Cisco Unified Border Element (SP Edition) supports SIP calls between endpoints using Transport Layer Security (TLS) for SIP signaling encryption and Secure Real-Time Protocol (SRTP) to provide RTP media encryption. However, these two encryption mechanisms may not be deployed simultaneously, depending on the required call flow invoked on the associated configuration.

Before delving further into SRTP passthrough configuration, it would be useful to understand the two concepts—the trusted vs. untrusted and encrypted vs. unencrypted.

The “trusted” implies that an associated adjacency is trusted to allow secure calls. Calls to a standard SIP: URI will be accepted. Calls to a secure SIPS: URI will be accepted and routed over a trusted adjacency (encrypted or unencrypted). The “untrusted” indicates that an associated adjacency is not trusted to carry secure calls. The calls to standard SIP: URI will be accepted. Calls to a secure SIPS: URI will be rejected immediately.

The “encrypted” implies that an associated adjacency uses TLS for SIP signaling and the “unencrypted” implies that an associated adjacency does not use TLS for SIP signaling.

The trusted/untrusted are configured in conjunction with encrypted/unencrypted as outlined in the following four (4) combinations. This is invoked using the security command:

- **untrusted-unencrypted**: The adjacency is untrusted and unencrypted. The adjacency is not trusted to carry secure SIP calls (calls with SIPS URI) and it does not use TLS encryption for SIP signaling.
- **untrusted-encrypted**: The adjacency is untrusted and encrypted. The adjacency is not trusted to carry secure SIP calls (calls with SIPS URI) and it does use TLS encryption for SIP signaling.
- **trusted-unencrypted**: The adjacency is trusted and unencrypted. The adjacency is trusted to carry secure SIP calls (calls with SIPS URI) and it does not use TLS encryption for SIP signaling.
- **trusted-encrypted**: The adjacency is trusted and encrypted. The adjacency is trusted to carry secure SIP calls (calls with SIPS URI) and it does use TLS encryption for SIP signaling.

When Cisco Unified Border Element (SP Edition) comes up, the default is to allow SRTP calls to pass through on the trusted interfaces.

The following are conditions of the SRTP Passthrough feature:

- SRTP Passthrough must be configured on both legs of the call. If the target adjacency does not support SRTP Passthrough, then the call is rejected by error message 415 (Unsupported Media Type).
- "m= .. RTP/SAVP .." and a="crypto:..." fields coming in on an Invite from one adjacency are passed on in an Invite to the target adjacency.
- “m= …RTP/SAVP…” is a required field in the Invite to trigger SRTP Passthrough behavior in the SBC.
The following shows a sample SRTP Invite and Response call flow from endpoints, as described in RFC-4568.

Offerer sends:

```
v=0
o=sam 2890844526 2890842807 IN IP4 10.47.16.5
s=SRTP Discussion
i=A discussion of Secure RTP
u=http://www.example.com/seminars/srtp.pdf
e=marge@example.com (Marge Simpson)
c=IN IP4 168.2.17.12
t=2873397496 2873404696
m=audio 49170 RTP/SAVP 0
a=crypto:1 AES_CM_128_HMAC_SHA1_80
  inline:WVNfX19zZWljdGwgKChgewkyMjA7fQp9CnVubGVz|2^20|1:4
  FEC_ORDER=FEC_SRTP
a=crypto:2 F8_128_HMAC_SHA1_80
  inline:MTIzNDU2Nzg5QUJDREUwMTIzNDU2Nzg5QUJjZGVm|2^20|1:4;
  inline:QUJjZGVmMTIzNDU2Nzg5QUJDREUwMTIzNDU2Nzg5|2^20|2:4
  FEC_ORDER=FEC_SRTP
```

Answerer replies:

```
v=0
o=jill 25690844 8070842634 IN IP4 10.47.16.5
s=SRTP Discussion
i=A discussion of Secure RTP
u=http://www.example.com/seminars/srtp.pdf
e=homer@example.com (Homer Simpson)
c=IN IP4 168.2.17.11
t=2873397526 2873405696
m=audio 32640 RTP/SAVP 0
a=crypto:1 AES_CM_128_HMAC_SHA1_80
  inline:PS1uQCveeCFCanVmcjkpPywjNWhcYD0mXXtxaVBR|2^20|1:4
```
Figure 38-1 diagram illustrates an SRTP Passthrough Call Flow.

Figure 38-1   SRTP Passthrough Call Flow

The SRTP Passthrough feature defines a new Call Admission Control (CAC) entry variable, called “srtp transport,” in the admission control table. If you configure the “srtp transport” variable, then CAC policy has the option to set the policy for the adjacency to either “allowed,” “disallowed,” or “trust only.”

Calls using SRTP Passthrough are allowed on the adjacencies specified by the policy. Where there are conflicting policies, “disallowed” overrides “allowed” which overrides “trusted-only.” If you configure the CAC policy, but you do not define the “srtp transport” variable, then the CAC policy takes the default value of “trusted-only” and restricts the SRTP calls between trusted endpoints.

See the `srtp support` command which sets the adjacency CAC policy for more information. The no form of the command sets the “srtp support” variable to “trusted-only.” The `show sbe sbe cac-policy-set table entry` command is modified to display a “SRTP Transport” field and whether the policy for the adjacency is to allow, disallow, or trust only for SRTP Transport.

You can set the CAC policy to allow SRTP passthrough and allow configuration of certain security policing, such as the following:

- Preventing secure calls on a given adjacency
- Ensuring that all media sent over a given adjacency is secure
- Ensuring that secure streams are signaled over secure SIP adjacencies.
Information About SRTP to RTP Interworking and SRTP Passthrough

Secure Real-time Transport Protocol (SRTP) to Real-time Transport Protocol (RTP) interworking is supported on Session Border Controller (SBC) services on Cisco ASR 1000 Series Aggregation Services Routers.

System Administrators may configure SRTP to RTP interworking to enable their networks to communicate with other networks and add additional security to a network. SRTP to RTP interworking allows networks that use SRTP to accept calls from networks that use RTP.

The SRTP to RTP interworking feature provides SBC with the ability to encrypt and decrypt data streams to and from both types of networks, SRTP networks and RTP networks.

SRTP to RTP interworking can be deployed on both User to Network Interfaces (UNI) and Network to Network Interfaces (NNI).

Features Supported

The following SRTP to RTP interworking features are supported by SBC:

- SBC-generated SRTP encryption and decryption keys.
- Configurable policies, for SRTP pass-through, termination, and re-origination when both caller and callee CAC policies support SRTP.
- SRTP to RTP interworking in distributed DBE mode via H.248.
- PD logs with information for verifying SBC call handling for different SRTP preference and policy settings. (Encryption keys are not displayed in PD logs.)
- Stateful Switchover (SSO) for SRTP streams.

CAC policies can support the following types of SRTP to RTP interworking:

- RTP-only
- SRTP-only
- SRTP-optional
- SRTP-prefer

When a CAC policy uses SRTP-only:

- All media streams associated to that CAC policy use SRTP. The SRTP stream is end-to-end if the peer adjacency supports SRTP. If the peer adjacency does not support SRTP, or if the policy configuration is set to terminate and re-originate, SBC performs the necessary SRTP encryption and decryption.
- SBC rejects incoming RTP calls and sends the appropriate response code.

When a CAC policy uses RTP-only:

- All media streams associated to that CAC policy use RTP. The RTP stream is end-to-end if the peer adjacency does not require SRTP. If the peer adjacency requires SRTP, SBC perform RTP to SRTP interworking.
- SBC rejects incoming SRTP calls and sends the appropriate response code.
When a CAC policy uses SRTP-optional:

- SRTP-optional is by negotiation on inbound calls.
- SBC accepts both incoming RTP and incoming SRTP calls.
- No RTP to SRTP interworking is needed for incoming RTP calls unless the callee CAC policy uses SRTP-only.
- No SRTP encryption is needed for incoming SRTP calls unless the callee CAC policy uses RTP-only, or the policy configuration prohibits pass-through mode.

When a CAC policy uses SRTP-prefer:

- SBC accepts either RTP or SRTP offers from endpoints.
- SBC offers SRTP to endpoints whether the inbound offer is RTP or SRTP.

The following SRTP and RTP statistics are collected and available in show commands at the global level and the adjacency level:

- Number of calls rejected due to RTP requested
- Number of calls rejected due to SRTP requested
- Number of calls using SRTP pass-through
- Number of calls performing RTP to SRTP interworking
- Number of calls using RTP
- Number of calls using SRTP

### SIP SRTP Offer Retry Feature

When the SIP SRTP Offer Retry feature is configured, using the `srtp {branch | callee | caller} retry rtp` command, and a 415 or 488 reject error code is generated in response to a prior SRTP (RTP/SAVP) offer, SBC reissues the offer, using RTP (RTP/AVP). This allows SBC to attempt to configure SRTP on a call leg and downgrade it to RTP if SRTP is not supported.

**Note**

415 and 488 error codes are general purpose errors. After the SRTP Offer Retry feature is configured, the SBC interprets that the 415 and 488 error codes are caused by an initial RTP/SAVP offer.

### Downgraded Response to an SRTP Offer

The `srtp {branch | callee | caller} response downgrade` command allows SBC to send an RTP/AVP answer in response to an RTP/SAVP offer and downgrade media security. For instance, if SRTP interworking is not configured in the CAC policy, and the caller offers RTP/SAVP, but the callee answers with RTP/AVP, this command allows the SBC to downgrade the answer to RTP/AVP instead of rejecting the call.

If downgrade is not set, SBC provides strict adherence to the offer/answer protocol and rejects RTP/SAVP offers that are not supported.

This is a non-standard procedure, and is not widely supported. SBC always supports receiving an SRTP downgrade answer, but only sends a downgrade answer when this downgrade flag is set.

Both of the following cases, for SRTP fallback to RTP, are subject to the overall per-side SRTP policy and RTP-SRTP interworking policy:
Information About SRTP to RTP Interworking and SRTP Passthrough

Chapter 38      Secure Media and SRTP Passthrough

• If the policy does not allow RTP at all, SBC does not attempt fallback.
• If the policy does not allow RTP-SRTP interworking, SBC allows a fallback on the answer side, but only if SBC can downgrade the offer side as well.

How SBC Processes SRTP

SRTP policies behave differently depending on how the following commands are set:
• srtp branch forbid | mandate | allow | prefer
• srtp caller forbid | mandate | allow | prefer
• srtp callee forbid | mandate | allow | prefer
• srtp media interworking forbid | allow
• srtp interworking forbid | allow

The settings for these commands are defined as follows:
• forbid—SRTP is not supported on the caller side or the callee side of the call.
• mandate—SRTP is mandatory on the caller side or the callee side of the call.
• allow—SRTP is optional on the caller side or the callee side of the call.
• prefer—SRTP is preferred on this adjacency. Both RTP and SRTP are accepted inbound, but only SRTP is offered outbound. When the prefer option is set on the offer side of a call, it functions the same as allow. When the prefer option is set on the answer side of the call, and there is a choice between offering RTP or SRTP, SRTP is offered.

SRTP Policy Passthrough Tables

The following tables show the behavior of SBC based on the configuration of SRTP policies for each side of a call.

Table 38-1 shows how SBC selects the SRTP passthrough type for a stream offered as RTP when an SRTP policy is present.

<table>
<thead>
<tr>
<th>SRTP Policy</th>
<th>SRTP Passthrough Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offer Side</td>
<td>Answer Side</td>
</tr>
<tr>
<td>Mandate</td>
<td>*</td>
</tr>
<tr>
<td>Forbid</td>
<td>Mandate</td>
</tr>
<tr>
<td>Forbid</td>
<td>Forbid</td>
</tr>
<tr>
<td>Forbid</td>
<td>Allow</td>
</tr>
<tr>
<td>Forbid</td>
<td>Prefer</td>
</tr>
<tr>
<td>Allow/Prefer</td>
<td>Mandate</td>
</tr>
<tr>
<td>Allow/Prefer</td>
<td>Forbid</td>
</tr>
<tr>
<td>Allow/Prefer</td>
<td>Allow</td>
</tr>
<tr>
<td>Allow/Prefer</td>
<td>Prefer</td>
</tr>
</tbody>
</table>
Table 38-2 shows how SBC selects the SRTP passthrough type for a stream offered as SRTP.

### Table 38-2 SBC Processing of SRTP Policies for SRTP Offers

<table>
<thead>
<tr>
<th>SRTP Policy</th>
<th>SRTP Passthrough Type</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Offer Side</strong></td>
<td><strong>Answer Side</strong></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Forbid</td>
<td>Mandate</td>
</tr>
<tr>
<td>Forbid</td>
<td>Forbid</td>
</tr>
<tr>
<td>Forbid</td>
<td>Allow</td>
</tr>
<tr>
<td>Forbid</td>
<td>Prefer</td>
</tr>
<tr>
<td>Mandate</td>
<td>Mandate</td>
</tr>
<tr>
<td>Mandate</td>
<td>Forbid</td>
</tr>
<tr>
<td>Mandate</td>
<td>Allow/Prefer</td>
</tr>
<tr>
<td>Allow/Prefer</td>
<td>Mandate</td>
</tr>
<tr>
<td>Allow/Prefer</td>
<td>Forbid</td>
</tr>
<tr>
<td>Allow/Prefer</td>
<td>Allow/Prefer</td>
</tr>
</tbody>
</table>

Table 38-3 shows how SBC selects the SRTP passthrough type when it receives a SIP 415 or SIP 488 rejection code in response to its SRTP offer, and Retry SRTP as RTP is set.

### Table 38-3 SBC Processing of SRTP Policies for SRTP Rejection with Retry SRTP as RTP

<table>
<thead>
<tr>
<th>SRTP Policy</th>
<th>SRTP Passthrough Type</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Offer Side</strong></td>
<td><strong>Answer Side</strong></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>*</td>
<td>Mandate</td>
</tr>
<tr>
<td>Mandate</td>
<td>Allow/Prefer</td>
</tr>
<tr>
<td>Allow/Prefer</td>
<td>Allow/Prefer</td>
</tr>
</tbody>
</table>

Table 38-4 shows how SBC selects the SRTP passthrough type when it receives an RTP downgrade answer to an SRTP offer.

### Table 38-4 SBC Processing of SRTP Policies for SRTP to RTP Downgrade Answer

<table>
<thead>
<tr>
<th>SRTP Policy</th>
<th>SRTP Passthrough Type</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Offer Side</strong></td>
<td><strong>Answer Side</strong></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>*</td>
<td>Mandate</td>
</tr>
<tr>
<td>Mandate</td>
<td>Allow/Prefer</td>
</tr>
<tr>
<td>Allow/Prefer</td>
<td>Allow/Prefer</td>
</tr>
</tbody>
</table>
Restrictions

The following restrictions apply to SRTP to RTP interworking and SRTP passthrough:

- Packet cable event messages continue to bill SRTP/RTP interworking calls and SRTP passthrough calls, and the billing does not indicate whether SRTP was used on one or both call legs.
- In late to early interworking and SRTP to RTP interworking, SBC does not support SRTP in a generated SDP offer. The call is forced to be an RTP-RTP call. If this violates the configured call policy, the event is logged and the call fails at setup.
- If a call has multiple streams (multiple m= lines in the SDP), each stream may have a different passthrough type. If any specific stream cannot be satisfied, the call is rejected. Calls with multiple streams and different passthrough types can occur when:
  - An offer is received containing a mix of RTP and SRTP streams.
  - An answer is downgraded from SRTP streams to a subset of RTP streams.
  - Some streams require interworking, others do not.
- SRTP capability is not signaled in H.248 and hence cannot be discovered automatically by SBC. This capability must be manually configured on SBC.
- SBC MG selection does not select an MG on the basis of which crypto-suites it supports.
- SBC does not allow the user to configure distinct SRTP session parameters on a per-call basis.
- SBC SRTP features do not work in conjunction with unsigned SRTP.
- SBC will fail an SRTP call if it receives a SIP forking answer.
- SIP late-early interworking does not support SRTP.
- H.323-SIP interworking does not support SRTP.
- SBC cannot terminate RFC5027 security preconditions signaling in RTP-SRTP calls.
- SBC does not support local call transfer of SRTP calls.
- SBC currently only supports the AES_CM_128_HMAC_SHA1_32 crypto suite.
- SBC does not refresh any master keys that it generates.
- SBC does not renegotiate master key rotation when the packet usage count is reached (as specified in RFC3711).
- If the transcoder does not support SRTP (such as MGX), SBC does not allow an SRTP-SRTP call. SBC cannot perform SRTP-RTP interworking on the two media gates on either side of the transcoder.
- RTP-SRTP and SRTP-RTP calls can be transcoded by a third-party transcoder. In such cases, the media through the transcoder RTP, and the interworking is performed by SBC on the side closest to the SRTP endpoint.

To configure SRTP to RTP interworking, see the Configuring CAC Policies for SRTP to RTP Interworking? section on page 38-23 and the CAC Policies for SRTP to RTP Interworking Configuration: Example? section on page 38-33.

You can display policy failure statistics for a specified source adjacency, using this existing command that has been updated for SRTP:

```
show sbc sbe call-stats src-adjacency
```

You can display all the calls on the SBEs, using this existing command that has been updated for SRTP:

```
show sbc sbe calls srtp-iw
```
Configuring Secure Media—Global Level

Note

The Unsignaled Secure Media feature was introduced in Cisco IOS XE Release 2.6 to allow configuration of secure media at a granular level using CAC table entry commands. With the introduction of this feature, the Configuring Secure Media—Global Level feature has been deprecated. If you are upgrading from a release earlier than Release 2.6, see the procedure described in the Configuring Unsignaled Secure Media at a Granular Level section on page 38-13.

Perform the following steps to configure secure media globally.

SUMMARY STEPS

1. configure terminal
2. sbc sbc-name
3. sbe
4. secure-media
5. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc sbc-name</td>
<td>Creates the SBC service on Cisco Unified Border Element (SP Edition) and enters into SBC configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# sbc mysbc</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Enters the mode of the signaling border element (SBE) function of the SBC.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> secure-media</td>
<td>Configures the SBC to treat every media flow as an encrypted media flow. This allows media packets, such as DTLS and SRTP packets, to pass through the SBC.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe)# secure-media</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> end</td>
<td>Exits SBE mode and returns to Privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe)# end</td>
<td></td>
</tr>
</tbody>
</table>
Configuring Unsigned Secure Media at a Granular Level

Use the following steps to configure both adjacencies and both call legs using CAC policy set to enable Unsigned Secure Media at a granular level.

Note

The caller and callee commands have been used in this procedure. In some scenarios, the branch command can be used as an alternative to the caller and callee command pair. The branch command has been introduced in Release 3.5.0. See the Configuring Directed Nonlimiting CAC Policies? section on page 7-37 for information about this command.

SUMMARY STEPS

1. configure terminal
2. sbc sbc-name
3. sbe
4. adjacency {sip | h323} adjacency-name
5. security [untrusted | trusted-encrypted | untrusted-encrypted | trusted-unencrypted]
6. exit
7. adjacency {sip | h323} adjacency-name
8. security [untrusted | trusted-encrypted | untrusted-encrypted | trusted-unencrypted]
9. exit
10. cac-policy-set policy-set-id
11. first-cac-table table-name
12. cac-table table-name
13. table-type limit list of limit tables
14. entry entry-id
15. match-value key
16. srtp support [allow | disallow | trusted-only]
17. caller secure-media
18. callee secure-media
19. action { cac-complete | next-table goto-table-name }
20. exit
21. complete
22. exit
23. active-cac-policy-set policy-set-id
24. end
25. show sbc sbc-name sbe cac-policy-set [id [table name [entry id]] | active [table name [entry id ]]] [detail]
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><code>configure terminal</code></td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> <code>Router# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td><code>sbc sbc-name</code></td>
<td>Creates the SBC service on Cisco Unified Border Element (SP Edition) and enters into SBC configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> <code>Router(config)# sbc mysbc</code></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td><code>sbe</code></td>
<td>Enters the mode of the signaling border element (SBE) function of the SBC.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> <code>Router(config-sbc)# sbe</code></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>`adjacency {sip</td>
<td>h323} adjacency-name`</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> <code>Router(config-sbc-sbe)# adjacency sip client</code></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>`security [untrusted</td>
<td>trusted-encrypted</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> <code>Router(config-sbc-sbe-adj-sip)# security trusted-encrypted</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> If this adjacency is <em>untrusted</em>, skip steps Step 4 through Step 6. You need to configure for an untrusted adjacency in a CAC policy table.</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td><code>exit</code></td>
<td>Exits the SBE SIP adjacency mode to the SBE mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> <code>Router(config-sbc-sbe-adj-sip)# exit</code></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>`adjacency {sip</td>
<td>h323} adjacency-name`</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> <code>Router(config-sbc-sbe)# adjacency sip server</code></td>
<td></td>
</tr>
</tbody>
</table>
## Chapter 38  Secure Media and SRTP Passthrough

### Configuring Unsignaled Secure Media at a Granular Level

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 8</td>
<td>security</td>
<td>Configures transport-level security (TLS) on a SIP adjacency. For granular-level Secure Media, configure the trusted adjacency as trusted-encrypted or trusted-unencrypted. Trusted means the adjacency is trusted to carry secure SIP calls (calls with SIPS URI). Encrypted means the adjacency uses TLS encryption for SIP signaling. Unencrypted means it does not use TLS encryption for SIP signaling.</td>
</tr>
<tr>
<td></td>
<td>[untrusted</td>
<td>If this adjacency is untrusted, skip steps Step 7 through Step 9. You need to configure for an untrusted adjacency in a CAC policy table.</td>
</tr>
<tr>
<td></td>
<td>trusted-encrypted</td>
<td></td>
</tr>
<tr>
<td></td>
<td>untrusted-encrypted</td>
<td></td>
</tr>
<tr>
<td></td>
<td>trusted-unencrypted</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-adj-sip)# security trusted-unencrypted</td>
<td></td>
</tr>
<tr>
<td>Step 9</td>
<td>exit</td>
<td>Exits the SBE SIP adjacency mode to the SBE mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-adj-sip)# exit</td>
<td></td>
</tr>
<tr>
<td>Step 10</td>
<td>cac-policy-set policy-set-id</td>
<td>Enters the mode of CAC policy set configuration within an SBE entity, creating a new policy set if necessary.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe)# cac-policy-set 1</td>
<td></td>
</tr>
<tr>
<td>Step 11</td>
<td>first-cac-table table-name</td>
<td>Configures the name of the first policy table to process. A CAC policy may have many tables configured. To start the application of the CAC policy, the first table that is used needs to be defined.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-cacpolicy)# first-cac-table testSecure</td>
<td></td>
</tr>
<tr>
<td>Step 12</td>
<td>cac-table table-name</td>
<td>Enters the mode for configuration of an admission control table (creating one if necessary) within the context of an SBE policy set.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-cacpolicy)# cac-table testSecure</td>
<td></td>
</tr>
</tbody>
</table>

### Note

- Policy set-id—Integer chosen by the user to identify the policy set. The range is 1 - 2147483647.
### Step 13

**Command or Action**  
`table-type limit list of limit tables`

**Example:**  
Router(config-sbc-sbe-cacpolicy-cactable)#
`table-type limit all`

**Purpose**  
Configures a new CAC Limit table type where you enter the criteria that is used to match the entries. 

*list of limit tables* can be one of the following values:

- **account**—Compare the name of the account.
- **adj-group**—Compare the name of the adjacency group.
- **adjacency**—Compare the name of the adjacency.
- **all**—No comparison type. All events match this type.
- **call-priority**—Compare with call priority.
- **category**—Compare the number analysis assigned category.
- **dst-account**—Compare the name of the destination account.
- **dst-adj-group**—Compare the name of the destination adjacency group.
- **dst-adjacency**—Compare the name of the destination adjacency.
- **dst-prefix**—Compare the beginning of the dialed digit string.
- **event-type**—Compare with CAC policy event types.
- **src-account**—Compare the name of the source account.
- **src-adj-group**—Compare the name of the source adjacency group.
- **src-adjacency**—Compare the name of the source adjacency.
- **src-prefix**—Compare the beginning of the calling number string.

### Step 14

**Command or Action**  
`entry entry-id`

**Example:**  
Router(config-sbc-sbe-cacpolicy-cactable)#
`entry 1`

**Purpose**  
Enters the mode to modify an entry in an admission control table. 

*entry-id*—Specifies the table entry.

### Step 15

**Command or Action**  
`match-value key`

**Example:**  
Router(config-sbc-sbe-cacpolicy-cactable-entry)#
`match-value call-update`

**Purpose**  
Configures the match-value of an entry in a CAC Limit table type.
### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>`srtp support [allow</td>
<td>disallow</td>
</tr>
</tbody>
</table>

Example:

```
Router(config-sbc-sbe-cacpolicy-cactable-entry)
# srtp support allow
```

If an adjacency is `untrusted` and you want granular-level Secure Media, you need to configure this step—configuring with `srtp support allow` will allow an SRTP call on the untrusted adjacency where the CAC policy is applied. Continue on to **Step 17**.

Configures the `srtp support` variable in the CAC table to allow or disallow SRTP Passthrough of secure media on the adjacency where the policy is applied.

- **allow**—allows SRTP Transport when an event matches this CAC policy.
- **disallow**—do not allow SRTP Transport when an event matches this CAC policy.
- **trusted-only**—allows SRTP Transport on a trusted adjacency (default) when an event matches this CAC policy.

Calls using SRTP Passthrough are allowed on the adjacencies specified by the policy.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td><code>caller secure-media</code></td>
<td>Configures a Secure Media call on the caller side.</td>
</tr>
</tbody>
</table>

Example:

```
Router(config-sbc-sbe-cacpolicy-cactable-entry)
# caller secure-media
```

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td><code>callee secure-media</code></td>
<td>Configures a Secure Media call on the callee side.</td>
</tr>
</tbody>
</table>

Example:

```
Router(config-sbc-sbe-cacpolicy-cactable-entry)
# callee secure-media
```

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>19</td>
<td>`action [cac-complete</td>
<td>next-table goto-table-name]`</td>
</tr>
</tbody>
</table>

Example:

```
Router(config-sbc-sbe-cacpolicy-cactable-entry)
# action cac-complete
```

- **cac-complete**—When an event matches, this CAC policy is complete.
- **next-table**—Specifies the name of the next cac table.
- **goto-table-name**—Specifies the table name identifying the next CAC table to process (or cac-complete, if processing should stop).

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td><code>exit</code></td>
<td>Exits CAC Table Entry mode and enters CAC Policy-set configuration mode.</td>
</tr>
</tbody>
</table>

Example:

```
Router(config-sbc-sbe-cacpolicy-cactable-entry)
# exit
```
Chapter 38 Secure Media and SRTP Passthrough

Configuring SRTP Passthrough

These steps show how to configure the CAC policy set to allow SRTP Passthrough.

**SUMMARY STEPS**

1. configure terminal
2. sbc sbc-name
3. sbe
4. cac-policy-set policy-set-id
5. first-cac-scope scope-name
6. first-cac-table table-name
7. cac-table table-name
8. table-type limit list of limit tables
9. entry entry-id
10. match-value key
11. srtp support [allow | disallow | trusted-only]
12. action [cac-complete | next-table | goto-table-name ]

**Command or Action** | **Purpose**
--- | ---
Step 21 complete | Completes the CAC-policy set after committing the full set.  
**Example:**  
Router(config-sbc-sbe-cacpolicy)# complete

Step 22 exit | Exits CAC Policy-set configuration mode and enters SBE mode.  
**Example:**  
Router(config-sbc-sbe-cacpolicy)# exit

Step 23 active-cac-policy-set policy-set-id | Sets the newly created CAC policy to be active. When the policy is active, it can no longer be modified.  
**Example:**  
Router(config-sbc-sbe)# active-cac-policy-set 1  
*policy-set-id*—Identifies the policy set that is made active.  
Range is 1 to 2147483647.

Step 24 end | Exits the SBE mode and returns to Privileged EXEC mode.  
**Example:**  
Router(config-sbc-sbe)# end

Step 25 show sbc name sbe cac-policy-set [id |table name [entry id]] active [table name [entry id]] [detail] | Displays detailed information for a given entry in a CAC policy table. In this example, that includes the caller/callee unsignaled secure media: Allowed fields and the security trusted-unencrypted for both adjacencies of the Secure Media call.  
**Example:**  
Router# show sbc mysbc sbe cac-policy-set 1 detail
13. exit
14. exit
15. complete
16. exit
17. active-cac-policy-set policy-set-id
18. end
19. show sbc sbc-name sbe cac-policy-set id table name entry entry

## Detailed Steps

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enables global configuration mode.</td>
</tr>
</tbody>
</table>
| **Example:**
   Router# configure terminal | |
| **Step 2** sbc sbc-name | Creates the SBC service on Cisco Unified Border Element (SP Edition) and enters into SBC configuration mode. |
| **Example:**
   Router(config)# sbc mysbc | |
| **Step 3** sbe | Enters the mode of the signaling border element (SBE) function of the SBC. |
| **Example:**
   Router(config-sbc)# sbe | |
| **Step 4** cac-policy-set policy-set-id | Enters the mode of CAC policy set configuration within an SBE entity, creating a new policy set if necessary. |
| **Example:**
   Router(config-sbc-sbe)# cac-policy-set 1 | *policy-set-id*—Integer chosen by the user to identify the policy set. The range is 1 - 2147483647.
### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| Step 5 | `first-cac-scope scope-name` | Configures scope at which limits should be initially defined when performing the admission control stage of the policy. Each CAC policy has a scope that is applied to it. This CAC policy applies on a per call basis. `scope-name` has one of the following values:  
  - **adj-group**—Limits for events from members of the same adjacency group.  
  - **call**—Limits are per single call.  
  - **category**—Limits per category.  
  - **dst-account**—Limits for events sent to the same account.  
  - **dst-adj-group**—Limits for events sent to the same adjacency group.  
  - **dst-adjacency**—Limits for events sent to the same adjacency.  
  - **dst-number**—Limits for events that have the same adjacency number.  
  - **global**—Limits are global (May not be combined with any other option).  
  - **src-account**—Limits for events from the same account.  
  - **src-adj-group**—Limits for events from the same adjacency group.  
  - **arc-adjacency**—Limits for events from the same adjacency.  
  - **src-number**—Limits for events that have the same source number. |
| Example: | `Router(config-sbc-sbe-cacpolicy)# first-cac-scope call` |
| Step 6 | `first-cac-table table-name` | Configures the name of the first policy table to process. A CAC policy may have many tables configured. To start the application of the CAC policy, the first table that is used needs to be defined. `table-name`—The admission control table that should be processed first. |
| Example: | `Router(config-sbc-sbe-cacpolicy)# first-cac-table testSecure` |
| Step 7 | `cac-table table-name` | Enters the mode for configuration of an admission control table (creating one if necessary) within the context of an SBE policy set. `table-name`—Name of the admission control table. |
| Example: | `Router(config-sbc-sbe-cacpolicy)# cac-table testSecure` |
### Command or Action

**Step 8**

```verbatim
table-type limit list of limit tables
```

**Example:**

Router(config-sbc-sbe-cacpolicy-cactable)#
table-type limit all

### Command or Action

**Step 9**

```verbatim
entry entry-id
```

**Example:**

Router(config-sbc-sbe-cacpolicy-cactable)#
entry 1

### Command or Action

**Step 10**

```verbatim
match-value key
```

**Example:**

Router(config-sbc-sbe-cacpolicy-cactable-entry)#
# match-value call-update

---

### Purpose

Configures a new CAC Limit table type where you enter the criteria that is used to match the entries.

*list of limit tables* can be one of the following values:

- **account**—Compare the name of the account.
- **adj-group**—Compare the name of the adjacency group.
- **adjacency**—Compare the name of the adjacency.
- **all**—No comparison type. All events match this type.
- **call-priority**—Compare with call priority.
- **category**—Compare the number analysis assigned category.
- **dst-account**—Compare the name of the destination account.
- **dst-adj-group**—Compare the name of the destination adjacency group.
- **dst-adjacency**—Compare the name of the destination adjacency.
- **dst-prefix**—Compare the beginning of the dialed digit string.
- **event-type**—Compare with CAC policy event types.
- **src-account**—Compare the name of the source account.
- **src-adj-group**—Compare the name of the source adjacency group.
- **src-adjacency**—Compare the name of the source adjacency.
- **src-prefix**—Compare the beginning of the calling number string.

Enters the mode to modify an entry in an admission control table.

*entry-id*—Specifies the table entry.

Configures the match-value of an entry in a CAC Limit table type.
## Configuring SRTP Passthrough

**Step 11**
`srtp support [allow | disallow | trusted-only]`

**Example:**
```
Router(config-sbc-sbe-cacpolicy-cactable-entry)
# srtp support allow
```

Configures the srtp support variable in the CAC table to allow or disallow SRTP Passthrough of secure media on the adjacency where the policy is applied.
- **allow**—allows SRTP Transport when an event matches this CAC policy.
- **disallow**—do not allow SRTP Transport when an event matches this CAC policy.
- **trusted-only**—allows SRTP Transport on a trusted adjacency (default) when an event matches this CAC policy.

Calls using SRTP Passthrough are allowed on the adjacencies specified by the policy. Where there are conflicting policies, “disallowed” overrides “allowed” which overrides “trusted-only.”

**Step 12**
`action [cac-complete | next-table goto-table-name]`

**Example:**
```
Router(config-sbc-sbe-cacpolicy-cactable-entry)
# action cac-complete
```

Configures the action to perform after this entry in an admission control table. Each entry requires a match criteria and an action. The action is to accept the transport.

- **cac-complete**—When an event matches, this CAC policy is complete.
- **next-table**—Specifies the name of the next cac table.
- **goto-table-name**—Specifies the table name identifying the next CAC table to process (or cac-complete, if processing should stop).

**Step 13**
`exit`

**Example:**
```
Router(config-sbc-sbe-cacpolicy-cactable-entry)
# exit
```

Exits CAC table entry submode and enters into cacpolicy cactable mode.

**Step 14**
`exit`

**Example:**
```
Router(config-sbc-sbe-cacpolicy-cactable)# exit
```

Exits cacpolicy cactable submode and enters into cacpolicy mode.

**Step 15**
`complete`

**Example:**
```
Router(config-sbc-sbe-cacpolicy-cactable)# complete
```

Completes the CAC policy after all the entries within the CAC tables have been configured.

**Step 16**
`exit`

**Example:**
```
Router(config-sbc-sbe-cacpolicy)# exit
```

Exits the cacpolicy submode and enters into SBE mode.
Configuring CAC Policies for SRTP to RTP Interworking

Use the following procedure to configure the CAC policies for the caller side and the callee side of a call for SRTP to RTP interworking.

SUMMARY STEPS

1. configure terminal
2. sbc sbc-name
3. sbe
4. cac-policy-set policy-set-id
5. first-cac-table table-name
6. cac-table table-name
7. table-type limit list of limit tables
(repeat steps 8 through 14 as many times as needed)
8. entry entry-id
9. match-value key
10. srtp support allow
11. action next-table goto-table-name
12. srtp caller forbid | mandate | allow | prefer
13. srtp interworking forbid | allow
14. srtp media interworking forbid | allow

CAC Table for Caller Side of the Call

Step 17

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>active-cac-policy-set policy-set-id</td>
<td>Sets the newly created CAC policy to be active. When the policy is active, it can no longer be modified.</td>
</tr>
<tr>
<td>Example: Router(config-sbc-sbe)# active-cac-policy-set 1</td>
<td>policy-set-id—Identifies the policy set that is made active. Range is 1 to 2147483647.</td>
</tr>
</tbody>
</table>

Step 18

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>end</td>
<td>Exits the SBE mode and returns to Privileged EXEC mode.</td>
</tr>
<tr>
<td>Example: Router(config-sbc-sbe)# end</td>
<td></td>
</tr>
</tbody>
</table>

Step 19

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>show sbc sbc-name sbe cac-policy-set id table name entry entry</td>
<td>Displays detailed output, including a “SRTP Transport” field and whether the policy for the adjacency is to allow, disallow, or trust only for SRTP Transport.</td>
</tr>
<tr>
<td>Example: Router# show sbc mysbc sbe cac-policy-set 1 table testSecure entry 1</td>
<td></td>
</tr>
</tbody>
</table>
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>configure terminal</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>Router# configure terminal</strong></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><strong>sbc sbc-name</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>Router(config)# sbc SBC1</strong></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><strong>sbe</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>Router(config-sbc)# sbe</strong></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td><strong>cac-policy-set policy-set-id</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>Router(config-sbc-sbe)# cac-policy-set 44</strong></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td><strong>first-cac-table table-name</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>Router(config-sbc-sbe-cacpolicy)# first-cac-table 44</strong></td>
</tr>
</tbody>
</table>

**CAC Table for Caller Side of the Call**
### Configuring CAC Policies for SRTP to RTP Interworking

#### Step 6

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>cac-table table-name</code></td>
<td>Enters the mode for configuration of an admission control table (creating one if necessary) within the context of an SBE policy set.</td>
</tr>
</tbody>
</table>

**Example:**

```
Router(config-sbc-sbe-cacpolicy)# cac-table 44
```

- `table-name`—Name of the admission control table.

#### Step 7

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>table-type limit list of limit tables</code></td>
<td>Configures the limit of the table types to be matched by the <code>match-value</code> command. For this example, use the following table type:</td>
</tr>
</tbody>
</table>

**Example:**

```
Router(config-sbc-sbe-cacpolicy-cactable)# table-type limit src-adjacency
```

- `src-adjacency`—Compare the name of the source adjacency.

**Repeat steps 8 through 14 as many times as necessary to configure as many entries as needed.**

#### Step 8

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>entry entry-id</code></td>
<td>Enters the mode to modify an entry in an admission control table.</td>
</tr>
</tbody>
</table>

**Example:**

```
Router(config-sbc-sbe-cacpolicy-cactable)# entry 1
```

- `entry-id`—Specifies the table entry.

#### Step 9

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>match-value key</code></td>
<td>Configures the match-value of an entry in a Call Admission Control (CAC) Limit Table.</td>
</tr>
</tbody>
</table>

**Example:**

```
Router(config-sbc-sbe-cacpolicy-cactable-entry)# match-value A
```

- `key`—Specifies the keyword used to match events. The format of the key is determined by the table-type limit.

#### Step 10

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>srtp support allow</code></td>
<td>Configures SRTP support.</td>
</tr>
</tbody>
</table>

**Example:**

```
Router(config-sbc-sbe-cacpolicy-cactable-entry)# srtp support allow
```

#### Step 11

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>action next-table goto-table-name</code></td>
<td>Configures the action to take when this routing entry is chosen.</td>
</tr>
</tbody>
</table>

**Example:**

```
Router(config-sbc-sbe-cacpolicy-cactable-entry)# action next-table 45
```

- `goto-table-name`—Specifies the next routing table to process when an event matches the entry.

#### Step 12

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>`srtp caller forbid</td>
<td>mandate</td>
</tr>
</tbody>
</table>

**Example:**

```
Router(config-sbc-sbe-cacpolicy-cactable-entry)# srtp caller forbid
```

- `forbid`—SRTP is not supported on the caller side of the call.
- `mandate`—SRTP is mandatory on the caller side of the call.
- `allow`—SRTP is optional on the caller side of the call.
- `prefer`—SRTP is preferred on this adjacency. Both RTP and SRTP are accepted inbound, but only SRTP is offered outbound.
### Command or Action

<table>
<thead>
<tr>
<th>Step 13</th>
<th>srtp interworking forbid</th>
<th>allow</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-cacpolicy-cactable-entry) # srtp interworking allow</td>
<td></td>
</tr>
</tbody>
</table>

Configures SRTP to RTP interworking.
- **forbid**—Prohibits SRTP to RTP interworking on the call.
- **allow**—Allows SRTP to RTP interworking on the call.

<table>
<thead>
<tr>
<th>Step 14</th>
<th>srtp media interworking forbid</th>
<th>allow</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-cacpolicy-cactable-entry) # srtp media interworking allow</td>
<td></td>
</tr>
</tbody>
</table>

Configures SRTP to RTP media interworking.
- **forbid**—Prohibits SRTP to RTP media interworking on the call.
- **allow**—Allows SRTP to RTP media interworking on the call.

### CAC Table for Callee Side of the Call

<table>
<thead>
<tr>
<th>Step 15</th>
<th>cac-table table-name</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-cacpolicy)# cac-table 45</td>
</tr>
</tbody>
</table>

Enters the mode for configuration of an admission control table (creating one if necessary) within the context of an SBE policy set.
- **table-name**—Name of the admission control table.

<table>
<thead>
<tr>
<th>Step 16</th>
<th>table-type limit list of limit tables</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-cacpolicy-cactable)# table-type limit src-adjacency</td>
</tr>
</tbody>
</table>

Configures the limit of the table types to be matched by the **match-value** command. For this example, use the following table type:
- **src-adjacency**—Compare the name of the source adjacency.

*Repeat steps 17 through 23 as many times as necessary to configure as many entries as needed.*

<table>
<thead>
<tr>
<th>Step 17</th>
<th>entry entry-id</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-cacpolicy-cactable)# entry 1</td>
</tr>
</tbody>
</table>

Enters the mode to modify an entry in an admission control table.
- **entry-id**—Specifies the table entry.

<table>
<thead>
<tr>
<th>Step 18</th>
<th>match-value key</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-cacpolicy-cactable-entry)# match-value A</td>
</tr>
</tbody>
</table>

Configures the match-value of an entry in a Call Admission Control (CAC) Limit Table.
- **key**—Specifies the keyword used to match events. The format of the key is determined by the table-type limit.

<table>
<thead>
<tr>
<th>Step 19</th>
<th>srtp support allow</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-cacpolicy-cactable-entry)# srtp support allow</td>
</tr>
</tbody>
</table>

Configures SRTP support.

<table>
<thead>
<tr>
<th>Step 20</th>
<th>action next-table goto-table-name</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-cacpolicy-cactable-entry)# action next-table 45</td>
</tr>
</tbody>
</table>

Configures the action to take when this routing entry is selected.
- **goto-table-name**—Specifies the next routing table to process if the event matches the entry.
Chapter 38  Secure Media and SRTP Passthrough

Configuring CAC Policies for SRTP to RTP Interworking

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 21</strong></td>
<td></td>
</tr>
<tr>
<td>srtp callee forbid</td>
<td>Configures SRTP for the callee side of the call.</td>
</tr>
<tr>
<td>Allow:</td>
<td></td>
</tr>
<tr>
<td># srtp callee forbid</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-cacpolicy-cactable-entry)</td>
<td></td>
</tr>
<tr>
<td># srtp callee forbid</td>
<td></td>
</tr>
<tr>
<td>• forbid—SRTP is not supported on the callee side of the call.</td>
<td></td>
</tr>
<tr>
<td>• mandate—SRTP is mandatory on the callee side of the call.</td>
<td></td>
</tr>
<tr>
<td>• allow—SRTP is optional on the callee side of the call.</td>
<td></td>
</tr>
<tr>
<td>• prefer—SRTP is preferred on this adjacency. Both RTP and SRTP are accepted inbound, but only SRTP is offered outbound.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 22</strong></td>
<td></td>
</tr>
<tr>
<td>srtp interworking forbid</td>
<td>Configures SRTP to RTP interworking.</td>
</tr>
<tr>
<td>Allow:</td>
<td></td>
</tr>
<tr>
<td># srtp interworking allow</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-cacpolicy-cactable-entry)</td>
<td></td>
</tr>
<tr>
<td># srtp interworking allow</td>
<td></td>
</tr>
<tr>
<td>• forbid—Prohibits SRTP to RTP interworking on the call.</td>
<td></td>
</tr>
<tr>
<td>• allow—Allows SRTP to RTP interworking on the call.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 23</strong></td>
<td></td>
</tr>
<tr>
<td>srtp media interworking forbid</td>
<td>Configures SRTP to RTP media interworking.</td>
</tr>
<tr>
<td>Allow:</td>
<td></td>
</tr>
<tr>
<td># srtp media interworking allow</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-cacpolicy-cactable-entry)</td>
<td></td>
</tr>
<tr>
<td># srtp media interworking allow</td>
<td></td>
</tr>
<tr>
<td>• forbid—Prohibits SRTP to RTP media interworking on the call.</td>
<td></td>
</tr>
<tr>
<td>• allow—Allows SRTP to RTP media interworking on the call.</td>
<td></td>
</tr>
</tbody>
</table>

Issue the complete command only after all entries are configured.

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 24</strong></td>
<td></td>
</tr>
<tr>
<td>complete</td>
<td>Completes the CAC-policy after all entries are entered.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-cacpolicy-cactable-entry)</td>
<td></td>
</tr>
<tr>
<td># complete</td>
<td></td>
</tr>
<tr>
<td><strong>Step 25</strong></td>
<td></td>
</tr>
<tr>
<td>end</td>
<td>Exits configuration mode and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-cacpolicy-cactable-entry)</td>
<td></td>
</tr>
<tr>
<td># end</td>
<td></td>
</tr>
<tr>
<td><strong>Step 26</strong></td>
<td></td>
</tr>
<tr>
<td>show sbc name sbe cac-policy-set id detail</td>
<td>Displays detailed information for the given entry ID in a CAC policy table. In this case, it shows the default values for SRTP-RTP interworking. For example:</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# show sbc SBC1 sbe cac-policy-set 1 detail</td>
<td></td>
</tr>
<tr>
<td>Caller SRTP support:</td>
<td>Inherit (default)</td>
</tr>
<tr>
<td>Callee SRTP support:</td>
<td>Inherit (default)</td>
</tr>
<tr>
<td>SRTP Interworking:</td>
<td>Inherit (default)</td>
</tr>
<tr>
<td>SRTP media Interworking:</td>
<td>Inherit (default)</td>
</tr>
</tbody>
</table>
SRTP Support for RTCP Multiplexed with RTP

In earlier releases, the SBC could process incoming RTP and RTCP streams that were sent over separate UDP channels. From Release 3.4S, the SBC can also process RTCP streams multiplexed with RTP streams and sent over a single UDP channel. The SBC distinguishes between RTCP and RTP streams by examining the payload format of each stream. This also applies to SRTCP streams multiplexed with SRTP streams.

**Note**
RFC 5761 describes the multiplexing of RTCP streams with RTP streams. The same principle applies to SRTCP and SRTP.

This feature is an enhancement to the support for interworking of RTP-based and SRTP-based endpoints that are linked through the SBC. The Cisco TelePresence System is an example of an RTP-based endpoint, and Cisco Umi TelePresence is an example of an SRTP-based endpoint. With the introduction of this feature, the SBC processes RTCP streams multiplexed with RTP streams coming from the Cisco TelePresence System. In a similar manner, the SBC identifies and correctly processes SRTCP streams multiplexed with SRTP streams coming from Cisco Umi TelePresence.

By default, the detection of RTCP streams multiplexed with RTP streams is disabled in the SBC. You can enable this feature by performing the procedure described in the following section.

Configuring the Detection of RTCP Multiplexed with RTP

This task explains how to configure the detection of RTCP streams multiplexed with RTP streams.

**Note**
The same procedure can be used to configure the detection of SRTCP streams multiplexed with SRTP streams.

**SUMMARY STEPS**

1. `configure terminal`
2. `sbc sbc-name`
3. `sbe`
4. `rtcp-mux`
**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>configure terminal</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router# configure terminal</td>
</tr>
<tr>
<td></td>
<td>Enters the global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><strong>sbc sbc-name</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config)# sbc MySbc</td>
</tr>
<tr>
<td></td>
<td>Enters the SBC service mode.</td>
</tr>
<tr>
<td></td>
<td>• sbc-name—Name of the SBC.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><strong>sbe</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc)# sbe</td>
</tr>
<tr>
<td></td>
<td>Enters the SBE configuration mode.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td><strong>rtcp-mux</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe)# rtcp-mux</td>
</tr>
<tr>
<td></td>
<td>Enables the detection of RTCP streams multiplexed with RTP streams.</td>
</tr>
<tr>
<td></td>
<td>By default, this feature is disabled.</td>
</tr>
</tbody>
</table>

**SRTP Support for SSRC-Based Multiplexing**

An SBC endpoint such as the Cisco TelePresence System multiplexes RTP streams of the same type (audio or video) on a single UDP channel. It uses the 32-bit synchronization source (SSRC) field of RTP streams to differentiate between discrete RTP streams originating from a single source.

When an SRTP-based or RTP-based endpoint sends multiplexed streams over a single UDP channel, the channel contains multiple streams and each stream has its own SSRC field. In earlier releases, the SBC could support only a single SSRC field in a UDP channel. Therefore, the SBC could not support interworking of endpoints that sent multiplexed SRTP and RTP. From Release 3.4S, the SBC can process multiple SSRC fields in multiplexed SRTP or RTP streams. In combination with SRTP support for RTCP multiplexed with RTP, this feature enhances interworking of RTP-based and SRTP-based endpoints.

**Configuring Global Secure Media Example**

This section provides a sample configuration for the Secure Media Passthrough feature.

```
Router# configure
Router(config)# sbc mysbc
Router(config-sbc)# sbe
Router(config-sbc-sbe)# secure-media
Router(config-sbc-sbe)# end
```
ConfiguringUnsigned,Granular-LevelSecureMedia:Examples

ThefollowingconfigurationexampleshowshowtheclientandserverSIPadjacenciesareconfiguredas
"securitytrusted-unencrypted"andhowtheCACtableentry1isconfiguredforsecuremediaonboth
thecallerandcallee sides.

The**caller**and**callee**commands have been used in this procedure. In some scenarios, the**branch**
command can be used as an alternative to the**caller**and**callee**command pair. The**branch**command
has been introduced in Release 3.5.0. See the "Configuring Directed Nonlimiting CAC
Policies" section on page 7-37 for information about this command.

```plaintext
...  
cac-policy-set 2  
  first-cac-table 1  
  cac-table 1  
  table-type limit all  
  entry 1  
    match-value call-update  
    caller secure-media  
    callee secure-media  
    action cac-complete  
    exit  
  complete  
  exit  
active-cac-policy-set 2  

adjacency sip client  
  nat force-off  
  security trusted-unencrypted  
  signaling-address ipv4 10.10.100.110  
  signaling-port 9060  
  remote-address ipv4 10.10.100.10 255.255.255.255  
  signaling-peer 10.10.100.10  
  signaling-peer-port 9060  
  attach  

adjacency sip server  
  nat force-off  
  security trusted-unencrypted  
  signaling-address ipv4 10.10.100.110  
  signaling-port 9070  
  remote-address ipv4 10.10.100.10 255.255.255.255  
  signaling-peer 10.10.100.10  
  signaling-peer-port 9070  
  attach
```

The following example shows how to configure granular-level unsignaled secure media where an
adjacency is *untrusted* by using the **srtp support allow** command on the untrusted adjacency in a CAC
policy table:

```plaintext
...  
cac-policy-set 2  
  first-cac-table 1  
  cac-table 1  
  table-type limit all  
  entry 1  
    match-value call-update
```
srtp support allow
caller secure-media
callee secure-media
action cac-complete
exit
complete
exit
active-cac-policy-set 2

The following example lists detailed information pertaining to CAC policy set 2, and shows how secure media is configured on the caller and callee sides:

Router# show sbc asr sbe cac-policy-set 2 detail

SBC Service "asr"

CAC Policy Set 2
Active policy set: Yes
Description: Averaging period: 60 sec
First CAC table: 1
First CAC scope: global
First CAC prefix length: 4294967256

Table name: 1
Description: Table type: policy-set

Entry 1
CAC scope:
CAC scope prefix length: 0
Action: CAC complete
Number of calls rejected: 0
Max calls per scope: Unlimited
Max call rate per scope: Unlimited
Max in-call rate: Unlimited
Max out-call rate: Unlimited
Max reg. per scope: Unlimited
Max reg. rate per scope: Unlimited
Max channels per scope: Unlimited
Max call rate per scope: Unlimited
Max out-call rate: Unlimited
Max reg. per scope: Unlimited
Max reg. rate per scope: Unlimited
Max channels per scope: Unlimited
Max updates per scope: Unlimited
Early media: Allowed
Early media direction: Both
Early media timeout: None
Transcoder per scope: Allowed
Caller Bandwidth-Field: None
Caller Bandwidth-Field: None
Media bypass: Allowed
Renegotiate Strategy: Delta
Max bandwidth per scope: Unlimited
SRTP Transport: Trusted-Only (by default)
Caller hold setting: Standard
Callee hold setting: Standard
Caller privacy setting: Never hide
Callee privacy setting: Never hide
Caller voice QoS profile: Default
Callee voice QoS profile: Default
Caller video QoS profile: Default
Callee video QoS profile: Default
Caller sig QoS profile: Default
Callee sig QoS profile: Default
Caller inbound SDP policy: None
Callee inbound SDP policy: None
Caller outbound SDP policy: None
Callee outbound SDP policy: None
Caller media disabled:
Strip All Answer
Callee media disabled:
Strip All Offer
Caller unsignaled secure media: Allowed
Callee unsignaled secure media: Allowed
Caller tel-event payload type: Default
Callee tel-event payload type: Default
Media flag:
  Ignore bandwidth-fields (b=), Telephone Event Interworking
Restrict codecs to list: Default
Restrict caller codecs to list: Default
Restrict callee codecs to list: Default
Maximum Call Duration: Unlimited

The following example shows an excerpt of detailed information for the callee side SIP adjacency 'server' showing that security trusted-unencrypted is configured:

Router# show sbc asr sbe adjacencies server detail
SBC Service 'asr'
  Adjacency server (SIP)
    Status: Attached
    Security: Trusted-Unencrypted

Configuring SRTP Passthrough Example

The following shows a configuration where the “srtp transport” variable is set in the CAC policy set 1 table for an adjacency to allow SRTP Pass through:
sbc SBE-NODE2-SBE1
  sbe
cac-policy-set 1
  first-cac-scope global
  first-cac-table STANDARD-LIST-BY-ACCOUNT
  cac-table STANDARD-LIST-BY-ACCOUNT
  table-type limit dst-account
  entry 1
    media-bypass-forbid
    match-value SIP-CUSTOMER-1
    max-num-calls 100
    max-call-rate 20
    max-bandwidth 1000000 bps
    callee-privacy never
    srtp support allow
    action cac-complete
    exit
  entry 2
    match-value SIP-CUSTOMER-2
    max-num-calls 100
    max-call-rate 20
    max-bandwidth 1000000 bps
    transcode-deny
    max-reg 500
    action cac-complete
    exit
  exit
active-call-policy-set 1

The following example displays entries in table CAC1 for CAC policy set 100 and shows that the SRTP Transport variable has been set to allow SRTP Pass through on whichever adjacency the policy is applied:

Router# show sbc SBC1 sbe cac-policy-set 100 table CAC1 entry 1000
  SBC Service 'SBC1'
CAC Policies for SRTP to RTP Interworking Configuration: Example

The following example shows specific details of how to configure the CAC policies for the caller side and the callee side of a call for SRTP to RTP interworking. Multiple entries with specific settings are given.

Policy set 100 table CAC1 entry 1000
Match value src-adjacency
Action CAC policy complete
Max calls Unlimited
Max call rate 100
Max registrations Unlimited
Max reg. rate Unlimited
Max bandwidth Unlimited
Max channels Unlimited
Transcoder Allowed
Caller privacy setting Never hide
Callee privacy setting Never hide
Early media Allowed
Early media direction Both
Early media timeout 0
Restrict codecs to list default
Media bypass Allowed
Number of calls rejected by this entry 0
SRTP Transport Allowed
Figure 38-2 shows the adjacencies that are used by the match-value command in this example.

**Figure 38-2   Adjacencies A, B, C, and D for Example**

```
configure terminal
sbc SBC1
  sbe
  cac-policy-set 44
    first-cac-table 44
      cac-table 44
        table-type limit src-adjacency
          entry 1
            match-value A
            srtp support allow
            action next-table 45
            srtp caller forbid
            srtp interworking allow
            srtp media interworking allow
```
CAC Policies for SRTP to RTP Interworking Configuration: Example

entry 2
match-value B
srtp support allow
action next-table 45
srtp caller forbid
srtp interworking allow
srtp media interworking allow

entry 3
match-value C
srtp support allow
action next-table 45
srtp caller mandate
srtp interworking allow
srtp media interworking allow

cac-table 45
table-type limit dst-adjacency

entry 1
match-value A
srtp support allow
action cac-complete
srtp callee forbid
srtp interworking allow
srtp media interworking allow

entry 2
match-value B
srtp support allow
action cac-complete
srtp callee forbid
srtp interworking allow
srtp media interworking allow

entry 3
match-value C
srtp support allow
action cac-complete
srtp callee mandate
srtp interworking allow
srtp media interworking allow

entry 4
match-value D
srtp support allow
action cac-complete
srtp callee mandate
srtp interworking allow
srtp media interworking allow

cac-table 45
table-type limit dst-adjacency

show sbc sbc1 sbe cac-policy-set 44 detail
CAC Policies for SRTP to RTP Interworking Configuration: Example
Implementing QoS (Marking)

Cisco Unified Border Element (SP Edition) supports quality of service (QoS) profiles that the integrator configures for IP packet marking on the data path. IP packet marking is used in Cisco Unified Border Element (SP Edition) in the following contexts:

- Configuring media packet real-time transport protocol (RTP) and real-time control protocol (RTCP) marking based on a per call scope.
- Supporting Differentiated Services Code Point (DSCP) marking as well as IP precedence/Type of Service (ToS) marking for voice service.
- Enabling the unique marking of media packets depending on the branch of the call (either the caller branch or the callee branch) on which the packets are sent.
- Supporting signaling and media packet marking based on Session Initiation Packet (SIP) resource priority header.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).

For a complete description of the commands used in this chapter, refer to the Cisco Unified Border Element (SP Edition) Command Reference: Unified Model at:


For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

Feature History for Implementing QoS (Marking)

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 2.4</td>
<td>This feature was introduced on the Cisco IOS XR along with support for the unified model.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 3.3S</td>
<td>The Flow Statistics Enhancements feature was introduced in the Cisco ASR 1000 Series Routers.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 3.4S</td>
<td>The QoS Demarcation feature was introduced in the Cisco ASR 1000 Series Routers. The Flow Statistics Enhancements feature mentioned in the preceding row is part of the QoS Demarcation feature.</td>
</tr>
</tbody>
</table>
Contents

This chapter contains the following sections:

- Prerequisites for Implementing QoS, page 38-2
- Information About Implementing QoS, page 38-2
- How to Implement QoS, page 38-2
- Implementing QoS Demarcation, page 38-10
- Configuration Examples of QoS Profiles, page 38-23

Prerequisites for Implementing QoS

The following is the prerequisite to implement QoS on Cisco Unified Border Element (SP Edition):

Information About Implementing QoS

To implement QoS marking on Cisco Unified Border Element (SP Edition), the user configures Cisco Unified Border Element (SP Edition) with a number of QoS profiles, which are given unique names to identify them. These QoS profiles are used exclusively for marking packets.

Each QoS profile contains the following mutually exclusive parameters.

- A 6-bit DSCP value to mark packets that match the QoS.
- A 3-bit IP precedence value and a 4-bit ToS value to mark packets that match the QoS.

Note

A default QoS profile that can be neither modified nor deleted is preconfigured on Cisco Unified Border Element (SP Edition). If the user does not define a QoS profile, the default QoS profile is used for marking packets.

QoS signaling profiles are currently supported only for SIP signaling.

How to Implement QoS

To implement QoS marking on Cisco Unified Border Element (SP Edition), follow the procedures in the following sections:

- Configuring QoS Profiles
- Selecting a QoS Profile Using CAC

Configuring QoS Profiles

This task configures a signaling QoS profile to use an IP precedence value of 1 and a ToS value of 12 to mark packets that match the QoS.
Note: QoS signaling profiles are currently supported only for SIP signaling.

## SUMMARY STEPS

1. configure terminal
2. `sbc `sbc-name`
3. sbe
4. `qos sig name`
5. `marking type`
6. `ip precedence value`
7. `ip tos value`

## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> <code>Router# configure terminal</code></td>
<td></td>
</tr>
</tbody>
</table>
| **Step 2** sbc `sbc-name` | Enters the mode of an SBC service.  
    - Use the `sbc-name` argument to define the name of the SBC. |
| **Example:** `Router(config)# sbc mySbc`  
  `Router(config-sbc)#` | |
| **Step 3** sbe | Enters the mode of an signaling border element (SBE) entity within a SBC service. |
| **Example:** `Router(config-sbc)# sbe`  
  `Router(config-sbc-sbe)#` | |
| **Step 4** `qos sig name` | Enters the mode of configuring a QoS profile. The `name` parameter must be the name of an existing QoS profile. The string “default” is reserved. |
| **Example:** `Router(config-sbc-sbe)# qos sig residential`  
  `Router(config-sbc-sbe-qos-sig)#` | |
| **Step 5** `marking type` | Configures whether the QoS policy marks packets with a DSCP value or an IP precedence and ToS value or a policy that does not mark. The `type` can be one of the following:  
    - dscp  
    - ip-precedence  
    - passthrough—creates a QoS policy that does not mark packets.  
  The `no` version of this command removes the QoS policy. |
| **Example:** `Router(config-sbc-sbe-qos-sig)# marking ip-precedence` | |
How to Implement QoS

Analyzing the SIP Resource-Priority Header

Users can configure Cisco Unified Border Element (SP Edition) to map SIP packets with Resource-Priority header strings to the following SBC priority values:

- Routine
- Priority
- Immediate
- Flash
- Flash override
- Critical

The Call Admission Control (CAC) uses the assigned priority value to choose the QoS profile.

The following task configures Cisco Unified Border Element (SP Edition) to assign priority value “flash” to a SIP packet with Resource-Priority header string “dsn.flash.”

### SUMMARY STEPS

1. `configure terminal`
2. `sbc service name`
3. `sbe`
4. `resource-priority-set name`
5. `resource-priority string value`
6. `priority priority-value`
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc sbc-name</td>
<td>Enters the mode of an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# sbc mySBC</td>
<td>- Use the <code>sbc-name</code> argument to define the name of the SBC.</td>
</tr>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Enters the mode of an SBE entity within a SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> resource-priority-set name</td>
<td>Enters the mode to map SIP Resource-Priority header string to SBC priority values.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe)# resource-priority-set dsn</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> resource-priority string value</td>
<td>Enters the mode to configure the priority of the Resource-Priority header string.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-rsrc-pri-set)# resource-priority dsn.flash</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> priority priority-value</td>
<td>Sets the SBC priority value of the Resource-Priority header string.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-rsrc-pri-set)# priority flash</td>
<td>The SBC priority value must be one of the following: - routine - priority - immediate - flash - flash-override - critical</td>
</tr>
</tbody>
</table>

## Configuring a Resource Priority Set on a SIP Adjacency

The following task configures the SIP adjacency “SipToIsp42” to use resource-priority-set “dsn.”

## SUMMARY STEPS

1. configure terminal
2. sbc service name
3. sbe
4. adjacency sip adjacency-name
5. `resource-priority-set name`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure terminal</td>
</tr>
<tr>
<td>Example:</td>
<td>Router# configure terminal</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>sbc sbc-name</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config)# sbc mysbc</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>sbe</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc)# sbe</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>adjacency sip adjacency-name</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe)# adjacency sip SipToIsp42</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>resource-priority-set name</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-adj-sip)# resource-priority-set dsn</td>
</tr>
</tbody>
</table>

### Selecting a QoS Profile Using CAC

This task configures calls from the account `cisco` to use the voice QoS profile `enterprise` for packets sent from Cisco Unified Border Element (SP Edition) to the original caller.

**Note**

This command can only be run at the per-call scope. The CAC policy does not get activated if this command is run at any other scope.

### SUMMARY STEPS

1. `configure terminal`
2. `sbc sbc-name`
3. `sbe`
4. `cac-policy-set policy-set-id`
5. `first-cac-scope scope-name`
6. `first-cac-table table-name`
7. `cac-table table-name`
8. `table-type limit list of limit tables`
9. `entry entry-id`
10. `match-value key`
11. `caller-voice-qos-profile profile-name`
12. `caller-video-qos-profile profile-name`
13. `caller-sig-qos-profile profile name`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><code>configure terminal</code> Enables global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router# configure terminal</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><code>sbc sbc-name</code> Enters the mode of an SBC service.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config)# sbc mysbc</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><code>sbe</code> Enters the mode of an SBE entity within an SBC service.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc)# sbe</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td><code>cac-policy-set policy-set-id</code> Enters the mode of Call Admission Control (CAC) policy set configuration within an SBE entity, creating a new policy set, if necessary.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe)# cac-policy-set 1</td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| 5    | `first-cac-scope scope-name` | Configures the scope at which to begin defining limits when performing the admission control stage of policy. The `scope-name` argument configures the scope at which limits should be initially defined. Possible values are:  
- `adj-group`  
- `call`  
- `dst-account`  
- `dst-adj-group`  
- `dst-adjacency`  
- `dst-number`  
- `global`  
- `src-account`  
- `src-adj-group`  
- `arc-adjacency` |
| 6    | `first-cac-table table-name` | Configures the name of the first policy table to process when performing the admission control stage of policy. |
| 7    | `cac-table table-name` | Enters the mode for configuration of an admission control table (creating one, if necessary) within the context of an SBE policy set. |

**Example:**

- `Router(config-sbc-sbe-cacpolicy)# first-cac-scope call`  
- `Router(config-sbc-sbe-cacpolicy)# first-cac-table MyCacTable`  
- `Router(config-sbc-sbe-cacpolicy)# cac-table MyCacTable`
### How to Implement QoS

#### Step 8

**Command or Action**

```bash
table-type limit list of limit tables
```

**Example:**

```bash
Router(config-sbc-sbe-cacpolicy-cactable)# table-type limit src-account
```

**Purpose**

Configures a CAC Limit table-type within the context of an SBE policy set.

*list of limit tables* can be one of the following values:

- **account**—Compare the name of the account.
- **adj-group**—Compare the name of the adjacency group.
- **adjacency**—Compare the name of the adjacency.
- **all**—No comparison type. All events match this type.
- **call-priority**—Compare with call priority.
- **category**—Compare the number analysis assigned category.
- **dst-account**—Compare the name of the destination account.
- **dst-adj-group**—Compare the name of the destination adjacency group.
- **dst-adjacency**—Compare the name of the destination adjacency.
- **dst-prefix**—Compare the beginning of the dialed digit string.
- **event-type**—Compare with CAC policy event types.
- **src-account**—Compare the name of the source account.
- **src-adj-group**—Compare the name of the source adjacency group.
- **src-adjacency**—Compare the name of the source adjacency.
- **src-prefix**—Compare the beginning of the calling number string.

#### Step 9

**Command or Action**

```bash
entry entry-id
```

**Example:**

```bash
Router(config-sbc-sbe-cacpolicy-cactable)# entry 1
```

**Purpose**

Enters the mode for configuring an entry in an admission control table, creating the entry, if necessary.

#### Step 10

**Command or Action**

```bash
match-value key
```

**Example:**

```bash
Router(config-sbc-sbe-cacpolicy-cac-table-ent)# match-value cisco
```

**Purpose**

Configures the match value of an entry in an admission control table.
Implementing QoS Demarcation

In the context of a network, a QoS demarcation point is a transit point within the network that provides features for measuring call quality and fixing problems that affect call quality. You can configure the SBC as a QoS demarcation point to meet the following objectives:

- Generate an alert when a problem in the network affects call quality.
- Provide information that can be used to determine the location of the problem.
- Calculate statistics that can assist in diagnosing and fixing the problem.

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 11</strong> caller-voice-qos-profile profile-name</td>
<td>Configures the QoS profile to use for voice media packets sent to the original caller.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-cacpolicy-cac-table-ent)# caller-voice-qos-profile enterprise</td>
<td></td>
</tr>
<tr>
<td><strong>Step 12</strong> caller-video-qos-profile profile-name</td>
<td>Configures the QoS profile to use for packets sent to the original caller.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-cacpolicy-cac-table-ent)# caller-video-qos-profile enterprise</td>
<td></td>
</tr>
<tr>
<td><strong>Step 13</strong> caller-sig-qos-profile profile-name</td>
<td>Configures the QoS profile to use for signaling packets sent to the original caller.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-cacpolicy-cac-table-ent)# caller-sig-qos-profile enterprise</td>
<td></td>
</tr>
</tbody>
</table>
The quality of a group of calls (or media streams processed by the SBC) can be determined by measuring parameters such as the following along the packet data path:

- Media packets lost while in transit from the sender to the receiver
- Media packets dropped from the set of media packets received
- Jitter in the media packets received
- Network latency in the media streams

Using these measurements, the SBC can calculate the following QoS statistics:

- Average local media packet jitter
  
  See RFC 3550 for the definition of local media packet jitter.

- Average remote media packet jitter
  
  See RFC 3550 for the definition of remote media packet jitter.

- Average round trip delay

- Mean Opinion Score for Conversational Quality, Estimated (MOS-CQE) score
  
  The MOS-CQE score provides an overall view of the statistics listed earlier. See Recommendations G.107 and G.113 of the ITU-T for the definition of the MOS-CQE score.

Note

The International Telecommunication Union (ITU) coordinates and assists in the development of telecommunications standards. The ITU Telecommunication Standardization Sector (ITU-T) is a division of the ITU. Recommendations G.107 and G.113 that are published by the ITU-T explain the MOS-CQE score and the method for calculating it. For more information about these recommendations, visit the ITU-T website at http://www.itu.int/ITU-T/index.html.

Of the various factors defined in Recommendations G.107 and G.113 for calculation of the MOS-CQE score, you can specify values for the following factors:

- Advantage (A) factor, which is specified at the per-adjacency level
- Equipment Impairment (Ie) factor, which is specified at the per-codec level
- Packet-Loss Robustness (Bpl) factor, which is specified at the per-codec level

Detailed information about these factors is available on the ITU-T website.

- Ratio of unanswered calls to the total number of calls

Note

The ratio of unanswered calls is not based on the measurements listed earlier.

- Ratio of media packets that are lost to the total number of media packets sent
- Ratio of media packets that are dropped to the total number of media packets received

Note

Stored QoS statistics data is lost after an RP failover.

For each statistic, you can configure a combination of the following alerts to denote the state of the statistic:

- Critical
- Major
Implementing QoS Demarcation

• Minor

For each alert, you specify a minimum (low) value and a maximum (upper) value. For statistics for which a higher value signifies an adverse effect on call quality, the alert changes when the upper limit of the earlier alert level is crossed. The following example illustrates how this works:

You specify the following alert levels for the Local Media Packet Jitter statistic:

• Major Low alert level: 60
• Major Upper alert level: 70
• Critical Low alert level: 71
• Critical Upper alert level: 80

A higher value of local media packet jitter indicates an adverse effect on call quality. While this statistic is being monitored, if the value of the statistic is increasing and crosses 80, the alert level changes to Critical. In contrast, if the value is decreasing and crosses 60, the alert changes to Normal. If the value is decreasing but does not go below 60, the alert level remains at Major.

Note

You can configure the SBC to generate an SNMP trap in response to changes in alert levels.

The reverse is true for the MOS-CQE score. If the alert levels listed earlier are specified for the MOS-CQE score, a MOS-CQE score higher than the specified Minor Upper alert level is classified as Normal. A value lower than the Critical Low alert level signifies that the statistic is in the Critical state.

You can specify the following time intervals at which a statistic must be measured. Because these are moving-average statistics, their values do not change suddenly over the boundaries of the time interval that you specify.

• Current 5 minutes—Statistics for the current 5-minute interval
• Current 15 minutes—Statistics for the current 15-minute interval
• Current hour—Statistics for the current 60-minute interval
• Current day—Statistics for the current day, starting from midnight
• Indefinitely—Statistics for the period starting from the last explicit reset

The following sections describe the procedures to configure the SBC for calculating the QoS statistics. Note that stored QoS statistics data is lost after an RP failover.

• Configuring the Calculation of the Local Jitter Ratio, page 38-12
• Configuring the G.107 Factors, page 38-15
• Configuring the Calculation of the MOS-CQE Score, page 38-18
• Configuring Alert Levels for the QoS Statistics, page 38-20
• Configuring SNMP Notifications for the QoS Statistics, page 38-23

Configuring the Calculation of the Local Jitter Ratio

Prior to Cisco IOS XE Release 3.3S, the Media Packet Forwarder (MPF) used the RTCP Sender Reports (SR) and Receiver Reports (RR) exchanged between a caller and a callee, and performed its own measurements on the media stream. QoS-related information was then passed to the media stream. Local jitter was not calculated because it requires tracking of the packet inter-arrival time, which is a processor-intensive operation in the MPF.
From Cisco IOS XE Release 3.3S, the SBC can be configured to calculate local jitter by tracking the percentage of calls that match criteria such as the source adjacency or destination adjacency. Local jitter is calculated according to the method specified in RFC 3550. The calculation is performed for both RTP streams and SRTP streams.

This task explains how to specify the percentage of calls for which the SBC must calculate the local jitter ratio. This task is one of the prerequisites for calculation of the MOS-CQE score.

**SUMMARY STEPS**

1. configure terminal
2. sbc sbc-name
3. sbe
4. adjacency {sip | h323} adjacency-name
5. local-jitter-ratio call-percentage
6. end
7. show sbc sbc-name sbe adjacencies adjacency-name detail

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure terminal</td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 2 sbc sbc-name</td>
<td>Enters the SBC service mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# sbc mySbc</td>
<td></td>
</tr>
<tr>
<td>Step 3 sbe</td>
<td>Enters the SBE entity mode within the SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td>Step 4 adjacency {sip</td>
<td>h323} adjacency-name</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe)# adjacency h323 adj1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• adjacency-name—Specifies the name of the SIP adjacency or H.323 adjacency.</td>
</tr>
</tbody>
</table>
## Implementing QoS Demarcation

### Step 5

**Command or Action**: `local-jitter-ratio call-percentage`  
**Purpose**: Specifies the percentage of calls that must be used to calculate the local jitter ratio.  
- **call-percentage**—Specifies the percentage of calls. The value is expressed as an integer in the range from 0 to 1000. For example, if you enter 205 as the value of `call-percentage`, the SBC uses 20.5 percent of the calls for measuring local jitter.  
The default value is 0 because jitter determination is a performance drain on the MPF. When the value is 0, the jitter ratio and MOS-CQE are not calculated for the adjacency.

**Example:**  
Router(config-sbc-sbe-adj-h323)#
local-jitter-ratio 205

### Step 6

**Command or Action**: `end`  
**Purpose**: Exits the SBE H.323 adjacency mode, and enters the privileged EXEC mode.

**Example:**  
Router(config-sbc-sbe-adj-h323)# end

### Step 7

**Command or Action**: `show sbc sbc-name sbe adjacencies adjacency-name detail`  
**Purpose**: Displays details of the specified adjacency. The output also includes the local jitter ratio.

**Example:**  
Router# show sbc mySbc sbe adjacencies h323adj detail

The following example displays details of an adjacency using the `show sbc sbe adjacencies detail` command. The output also includes the `call-percentage` parameter value.

Router# show sbc mySbc sbe adjacencies adj1 detail

---

**SBC Service "mySbc"**  
**Adjacency adj1 (H.323)**  
**Status:** Attached  
Signaling address: 1.0.0.3:1720 (default)  
Signaling-peer: 40.40.40.4:1720 (default)  
Admin Domain: None  
Account:  
Media passthrough: Yes  
Group:  
Hunting triggers: Global Triggers  
Hunting mode: Global Mode  
Technology Prefix:  
H245 Tunnelling: Enabled  
Fast-Slow Interworking: None  
Trust-level: Untrusted  
Call-security: Insecure  
Realm: None  
Warrant Match-Order: None  
Local Jitter Ratio: 205/1000  
Calc Moscqe: 0/1000  
G107A factor: 0  
H225 address block: Disabled (default)  
H225 address usage: h323id (default)
Configuring the G.107 Factors

The Advantage (A) factor, Equipment Impairment (Ie) factor, and Packet-Loss Robustness (Bpl) factor are used in the calculation of the MOS-CQE score. From Cisco IOS XE Release 3.4S, you can specify values for these factors.

This task explains how to configure the Advantage factor, Equipment Impairment factor, and Packet-Loss Robustness factor.

**SUMMARY STEPS**

1. configure terminal
2. sbc sbc-name
3. sbe
4. adjacency {sip | h323} adjacency-name
5. g107a-factor factor-number
6. exit
7. codec system sys-codec id payload-id
8. g107 ie factor-number
9. g107 bpl factor-number
10. end
11. show sbc sbc-name sbe adjacencies adjacency-name detail
12. show sbc sbc-name sbe codecs name codec-name

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td>Example: Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc sbc-name</td>
<td>Enters the SBC service mode.</td>
</tr>
<tr>
<td>Example: Router(config)# sbc mySbc</td>
<td>• sbc-name—Name of the SBC.</td>
</tr>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Enters the SBE entity mode within a SBC service.</td>
</tr>
<tr>
<td>Example: Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> adjacency {sip</td>
<td>h323} adjacency-name</td>
</tr>
<tr>
<td>Example: Router(config-sbc-sbe)# adjacency h323 adj1</td>
<td>• adjacency-name—Name of the SIP adjacency or H.323 adjacency.</td>
</tr>
</tbody>
</table>
### Chapter 38      Implementing QoS (Marking)

#### Implementing QoS Demarcation

**Step 5**

```bash
Router(config-sbc-sbe-adj-h323)# g107a-factor 10
```

Sets the Advantage factor.
- `factor-number`—Value of the Advantage factor. The range is from 0 to 20.

The default value is 0. See Recommendation G.107 for information about the significance of this default value.

**Step 6**

```bash
Router(config-sbc-sbe)# exit
```

Exits the SBE H.323 adjacency mode or the SBE SIP adjacency mode, and enters the SBE entity mode.

**Step 7**

```bash
Router(config-sbc-sbe)# codec system PCMU id 0
```

Enters the codec definition mode.

**Step 8**

```bash
Router(config-sbc-sbe-codec-def)# g107 ie 20
```

Sets the Equipment Impairment factor.
- `factor-number`—Value of the Equipment Impairment factor. The range is from 0 to 50.

See Appendix I of Recommendation G.113 for information about the values that you can set for various codecs. If you have a custom codec, you can set a value that best matches the impairment introduced by the codec.

The default value is 0. See Appendix I of Recommendation G.113 for information about the significance of this default.

**Step 9**

```bash
Router(config-sbc-sbe-codec-def)# g107 bpl 30
```

Sets the Packet-Loss Robustness factor.
- `factor-number`—Specifies the value of the Packet-Loss Robustness factor, which can range from 1 to 40.

See Appendix I of Recommendation G.113 for information about the values that you can set for various codecs. If you have a custom codec, you can set a value that best matches the Packet-Loss Robustness factor for the codec.

The default value is 1. See Appendix I of Recommendation G.113 for information about the significance of this default.

**Step 10**

```bash
Router(config-sbc-sbe-adj-h323)# end
```

Exits the codec definition mode, and enters the privileged EXEC mode.
Chapter 38  Implementing QoS (Marking)  

Implementing QoS Demarcation

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 11**  
show sbc sbc-name sbe adjacencies  
adjacency-name detail | Displays details of the specified adjacency. The output includes the value set for the Advantage factor. |

**Example:**
Router# show sbc mySbc sbe adjacencies h323adj detail

| **Step 12**  
show sbc sbc-name sbe codecs name codec-name | Displays details of the specified codec. The output includes the Equipment Impairment factor and Packet-Loss Robustness factor values. |

**Example:**
Router# show sbc mySbc sbe codecs name PCMU

The following example displays the details of a specified adjacency using the **show sbc sbc-name sbe adjacencies** detail command. The output includes the Advantage factor value.

Router# show sbc mySbc sbe adjacencies adj1 detail

SBC Service "mySbc"  
Adjacency adj1 (H.323)  
Status: Attached  
Signaling address: 1.0.0.3:1720 (default)  
Signaling-peer: 40.40.40.4:1720 (default)  
Admin Domain: None  
Account:  
Media passthrough: Yes  
Group:  
Hunting triggers: Global Triggers  
Hunting mode: Global Mode  
Technology Prefix:  
H245 Tunnelling: Enabled  
Fast-Slow Interworking: None  
Trust-level: Untrusted  
Call-security: Insecure  
Realm: None  
Warrant Match-Order: None  
Local Jitter Ratio: 1000/1000  
Calc Mosqe: 305/1000  
G107A factor: 10  
H225 address block: Disabled (default)  
H225 address usage: h323id (default)

The following example displays the details of a specified codec using the **show sbc sbc-name sbe codecs name codec** command. The output includes the Equipment Impairment factor and Packet-Loss Robustness factor values.

Router# show sbc mySbc sbe codecs name PCMU

codec_name = PCMU  
static_payload_id = 0  
codec_type = sample  
clock_rate = 8000  
packet_time = 10  
bandwidth = 64000  
sample_size = 8  
num_channels = 1  
max_fpp = 20  
media_type = audio  
g107 bpl = 40  
g107 ie = 50
Implementing QoS Demarcation

Chapter 38 Implementing QoS (Marking)

Implementing QoS Demarcation

options = transcode, inband-dtmf

Configuring the Calculation of the MOS-CQE Score

This section describes the procedure to configure a target MOS-CQE score.

The following are prerequisites for calculation of the MOS-CQE score:

- Performing the procedure described in the Configuring the Calculation of the Local Jitter Ratio section on page 38-12. Note that it is optional to configure calculation of the local jitter ratio. If you do not perform the procedure, the default value set for the percentage of calls for which the local jitter is to be calculated is used in the calculation of the MOS-CQE score.

- Performing the procedure described in the Configuring the G.107 Factors section on page 38-15. Note that it is optional to configure the G.107 factors. If you do not perform this procedure, the default values that are set for these factors are used to calculate the MOS-CQE score.

SUMMARY STEPS

1. configure terminal
2. sbc sbc-name
3. sbe
4. adjacency {sip | h323} adjacency-name
5. calc-moscqe call-percentage
6. end
7. show sbc sbc-name sbe adjacencies adjacency-name detail

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure terminal</td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td>Example: Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 2 sbc sbc-name</td>
<td>Enters the SBC service mode.</td>
</tr>
<tr>
<td>Example: Router(config)# sbc mySbc</td>
<td></td>
</tr>
<tr>
<td>Step 3 sbe</td>
<td>Enters the SBE entity mode within a SBC service.</td>
</tr>
<tr>
<td>Example: Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td>Step 4 adjacency {sip</td>
<td>h323} adjacency-name</td>
</tr>
<tr>
<td>Example: Router(config-sbc-sbe)# adjacency h323 adj1</td>
<td></td>
</tr>
</tbody>
</table>
## Chapter 38  Implementing QoS (Marking)

### Implementing QoS Demarcation

The following example displays the details of a specified adjacency using the `show sbc sbe adjacencies detail` command. The output also includes the `call-percentage` parameter value.

```
Router# show sbc mySbc sbe adjacencies adj1 detail

SBC Service "mySbc"
Adjacency adj1 (H.323)
  Status: Attached
  Signaling address: 1.0.0.3:1720 (default)
  Signaling-peer: 40.40.40.4:1720 (default)
  Admin Domain: None
  Media passthrough: Yes
  Group:
    Hunting triggers: Global Triggers
    Hunting mode: Global Mode
  Technology Prefix:
    H245 Tunnelling: Enabled
    Fast-Slow Interworking: None
    Trust-level: Untrusted
    Call-security: Insecure
    Realm: None
  Warrant Match-Order: None
  Local Jitter Ratio: 1000/1000
  Calc Moscqe: 305/1000
  G107A factor: 0
  H225 address block: Disabled (default)
  H225 address usage: h323id (default)
```

### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td><code>calc-moscqe call-percentage</code></td>
<td>Specifies the percentage of calls that must be used to calculate the MOS-CQE score.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Router(config-sbc-sbe-adj-h323)# calc-moscqe 305</code></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td><code>end</code></td>
<td>Exits the SBE SIP adjacency mode or SBE H.323 adjacency mode, and returns to the privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Router(config-sbc-sbe)# end</code></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td><code>show sbc sbc-name sbe adjacencies adjacency-name detail</code></td>
<td>Displays details of the specified adjacency. The output also includes the value that is set for the <code>call-percentage</code> parameter.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Router# show sbc mySbc sbe adjacencies h323adj detail</code></td>
<td></td>
</tr>
</tbody>
</table>
Implementing QoS Demarcation

You can display the MOS-CQE score calculated by the SBC by running the `show sbc sbe call-stats per-adjacency` command. A sample output of this command is provided later in this chapter.

## Configuring Alert Levels for the QoS Statistics

This task explains how to configure alert levels for the QoS statistics.

### SUMMARY STEPS

1. `configure terminal`
2. `sbc sbc-name`
3. `sbe`
4. `statistics {lcl-jit | mos-cqe | mpd-pct | mpl-pct | rmt-jit | rtd | ucr}`
5. `currenthour {adjacency adjacency-name {critical low value upper value | major low value upper value [critical low value upper value] | minor low value upper value [critical low value upper value] | [major low value upper value [critical low value upper value]] | default [critical low value upper value [major low value upper value [critical low value upper value]] [major low value upper value [critical low value upper value]]]}}`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td></td>
</tr>
<tr>
<td><code>configure terminal</code></td>
<td>Enables the global configuration mode.</td>
</tr>
</tbody>
</table>

**Example:**

Router# configure terminal

<table>
<thead>
<tr>
<th>Step 2</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><code>sbc sbc-name</code></td>
<td>Enters the SBC service mode.</td>
</tr>
</tbody>
</table>

**Example:**

Router(config)# sbc mySbc

- `sbc-name`—Name of the SBC.

### Note

Depending on your requirement, you can use `current15mins`, `currentday`, or `currentindefinite` instead of `currenthour`. The time interval for which statistical data is monitored varies according to the command that you run. The syntax is the same for all these commands. The time intervals for which these commands are applicable are described at the start of the "Implementing QoS Demarcation? section on page 38-10."
### Implementing QoS Demarcation

#### Step 3
**Command or Action:**
```
sbe
```
**Example:**
```
Router(config-sbc)# sbe
```
**Purpose:** Enters the SBE configuration mode.

#### Step 4
**Command or Action:**
```
statistics (lcl-jit | mos-cqe | mpd-pct | mpl-pct | rmt-jit | rtd | ucr)
```
**Example:**
```
Router(config-sbc-sbe)# statistics lcl-jit
```
**Purpose:** Specifies the statistic for which you want to set alert levels. You can specify one of the following statistics:
- **lcl-jit**—Specifies the average local media packet jitter.
- **mos-cqe**—Specifies the MOS-CQE score.
- **mpd-pct**—Specifies the ratio of media packets that are dropped to the total number of media packets received.
- **mpl-pct**—Specifies the ratio of media packets that are lost to the total number of media packets sent.
- **rmt-jit**—Specifies the average remote media packet jitter.
- **rtd**—Specifies the average round trip delay.
- **ucr**—Specifies the ratio of unanswered calls to the total number of calls.
Chapter 38      Implementing QoS (Marking)

Implementing QoS Demarcation

The following example displays the call statistics pertaining to the adjacency (since the last explicit reset) using the `show sbc sbe call-stats per-adjacency` command. The output also includes QoS statistics.

```
Router# show sbc Mysbc sbe call-stats per-adjacency adj1 currentindefinite

... Statistics for the current hour for adjacency adj1

Stats Reset Timestamp:
```

### Command or Action

**Step 5**

```
currenthour {adjacency adjacency-name (critical low value upper value | major low value upper value [critical low value upper value] | minor low value upper value [major low value upper value [critical low value upper value]] ) | default (critical low value upper value [major low value upper value] | minor low value upper value [major low value upper value [critical low value upper value]] )}
```

**Example:**

```
Router(config-sbc-sbe-stats)# currenthour default critical low 30 upper 50
```

**Purpose**

Specifies that statistical data must be monitored for the next hour for the QoS statistic specified by the `statistics` command.

**Note**

Depending on your requirement, you can use `current15mins, currentday, or currentindefinite` instead of `currenthour`. The time interval for which statistical data is monitored varies according to the command that you run. The syntax is the same for all these commands. The time intervals for these commands is described at the start of the `Implementing QoS Demarcation` section on page 38-10.

- **adjacency**—Specifies that alert levels must be set for the specified adjacency.
- **adjacency-name**—Name of the adjacency.
- **critical**—Specifies the lower and upper limits for the Critical alert level.
- **low**—Specifies the lower limit for the alert level.
- **value**—Value of the lower limit or upper limit.
- **upper**—Specifies the upper limit for the alert level.
- **major**—Specifies the lower limit and upper limit for the Major alert level.
- **minor**—Specifies the lower limit and upper limit for the Minor alert level.
- **default**—Specifies that alert levels must be set for all adjacencies on the SBC.

**Step 6**

```
end
```

**Example:**

```
Router(config-sbc-sbe-adj-h323)# end
```

**Purpose**

Exits the SBE entity mode, and enters the privileged EXEC mode.

**Step 7**

```
show sbc sbc-name sbe call-stats per-adjacency adjacency-name period
```

**Example:**

```
Router# show sbc mySbc sbe call-stats per-adjacency adj1 currentindefinite
```

**Purpose**

Displays QoS statistics for the specified adjacency.

- **sbc-name**—Name of the SBC.
- **adjacency-name**—Name of the adjacency.
- **period**—Interval for which the statistics must be displayed.

---

The following example displays the call statistics pertaining to the adjacency (since the last explicit reset) using the `show sbc sbe call-stats per-adjacency` command. The output also includes QoS statistics.

```
Router# show sbc Mysbc sbe call-stats per-adjacency adj1 currentindefinite

... Statistics for the current hour for adjacency adj1

Stats Reset Timestamp:
```
Chapter 38      Implementing QoS (Marking)

Configuration Examples of QoS Profiles

This section provides the following configuration examples:

- Configuring a QoS Voice Profile Using IP Precedence Marking: Example
- Configuring a QoS Voice Profile Using DSCP Marking: Example
- Choosing a QoS Profile Using CAC: Example
- Configuring a SIP Adjacency Using a Resource-Priority Set: Example

Configuring a QoS Voice Profile Using IP Precedence Marking: Example

This task configures a QoS voice profile to use an IP precedence value of 1 and a ToS value of 12 to mark packets that match the QoS.

```
configure
sbc mySBC
sbe
qos voice residential
marking ip-precedence
   ip precedence 1
   ip tos 12
```
Configuring a QoS Voice Profile Using DSCP Marking: Example

This task configures a QoS voice profile to mark packets with a DSCP value of 10.

```plaintext
configure
sbc mysbc
sbe
  qos voice residential
  marking dscp
  dscp 10
```

Choosing a QoS Profile Using CAC: Example

This task configures calls from the account “cisco” to use the voice QoS profile “enterprise” for packets sent from Cisco Unified Border Element (SP Edition) to the original caller.

```plaintext
configure
sbc mysbc
sbe
  cac-policy-set 1
    first-cac-scope call
    first-cac-table MyCacTable
    cac-table MyCacTable
    table-type limit src-account
    entry 1
    match-value cisco
    caller-voice-qos-profile enterprise
    caller-video-qos-profile enterprise

sbc mysbc
sbe
  cac-policy-set 1
    first-cac-scope call
    first-cac-table MyCacTable
    cac-table MyCacTable
    table-type limit src-account
    entry 1
    match-value cisco
    caller-video-qos-profile enterprise
    caller-voice-qos-profile enterprise
!
!
```

Configuring a SIP Adjacency Using a Resource-Priority Set: Example

The following example shows how to configure a SIP adjacency using a resource-priority set:

```plaintext
configure
sbc mysbc
sbe
  adjacency sip SipToIsp42
  resource-priority-set dsn
```
Implementing Transcoding

Transcoding is the process of translating a media stream encoded using one codec into a media stream encoded using another codec. For example, translating a media stream encoded as Pulse Code Modulation u-law (PCMU) into one encoded as ITU-T G.726-32.

The primary reason for transcoding configurations is to configure the capabilities of external media transcoding devices when these devices cannot be discovered automatically. In-band auto discovery of transcoder capabilities is currently not supported. Therefore, this step must be done when configuring all connections to all current remote transcoding devices.

Note

Transcoding configurations can be skipped altogether if the described reason does not apply.

Media gateways are allowed to connect whether or not configuration has been supplied for them. To help avoid configuration errors, the signaling border element (SBE) logs a warning if an incoming connection is received from a media gateway that is not a data border element (DBE) and does not have transcoding configured.

Note

The Transcoding feature is supported in the unified model for Cisco IOS XE Release 2.5 and later.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).

For a complete description of the commands used in this chapter, refer to the Cisco Unified Border Element (SP Edition) Command Reference: Unified Model at:


For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

Feature History for Implementing SBC Transcoding

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 2.5</td>
<td>This feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 3.3S</td>
<td>The Voice Transcoding Per Adjacency Statistics feature was added to the Cisco ASR 1000 Series Aggregation Services Routers.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 3.11S</td>
<td>The Blended Transcoding feature was added to the Cisco ASR 1000 Series Aggregation Services Routers.</td>
</tr>
</tbody>
</table>
Contents

This chapter contains the following sections:

- Prerequisites for Implementing Transcoding, page 40-2
- Restrictions for Implementing Transcoding, page 40-2
- Restrictions for Media Gateway-Assisted DMTF Interworking, page 40-3
- Information About Transcoding, page 40-3
- Configuration Examples for Implementing Transcoding, page 40-13
- Verification, page 40-15
- Voice Transcoding Per Adjacency Statistics, page 40-16
- Configuring the Voice Transcoding Per Adjacency Statistics, page 40-16
- Media Gateway-Assisted DTMF Interworking, page 40-19
- Blended Transcoding, page 40-27

Prerequisites for Implementing Transcoding

The following prerequisites are required to implement SBC transcoding:

- Before implementing these features, Cisco Unified Border Element (SP Edition) must already be configured.
- All SBE and DBE configurations required to make simple calls must already be configured. Transcoding configurations follow these configurations.

Restrictions for Implementing Transcoding

The following are restrictions of the Implementing Transcoding feature:

- The H.323 fast-start calls will be dropped to slow-start procedure if transcoding is required. This can be achieved by the callee side rejecting the fast-start request.
- No transcoding support for H.323 to SIP interworked calls.
- No transcoding support for H.323 to H.323 interworked calls.
- The only codecs supported for H.323 transcoding are G.711 (PCMU and PCMA) and G.729 (with and without annex B).
- When audio transcoding is in operation, the SBC does not support sending and receiving RFC 2833 in-band packets to and from the SBC and interworking RFC 2833 packets with out-of-band SIP INFO or SIP NOTIFY Relay messages on the other call leg.

The following are DTMF interworking restrictions when transcoding is used:

- Signaling and media DTMF interworking is not supported when transcoding is performed on the call
When audio transcoding is in operation, the SBC does not support sending and receiving RFC 2833 in-band packets to and from the SBC and interworking RFC 2833 packets with out-of-band SIP INFO or SIP NOTIFY Relay messages on the other call leg.

Restrictions for Media Gateway-Assisted DMTF Interworking

Following are the restrictions of the Media Gateway-Assisted DMTF interworking feature:

- The SBC supports the use of transcoders, such as a Cisco MGX 8880, only in SIP-SIP calls. DTMF interworking with transcoders are not supported for H.323 calls.
- The SBC cannot interwork DTMF with transcoders that cannot pass through DTMF.
- When a Cisco MGX 8880 is not used as transcoders, only SIP-SIP calls are supported.

Information About Transcoding

Transcoding is the process of translating a media stream encoded using one codec into a media stream encoded using another codec. For example, translating a media stream encoded as PCMU into one encoded as G.726-32.

Transcoding is supported using external digital signal processor (DSP) hardware. A Cisco MGX 8880 Media Gateway can be used to provide the transcoding function for one or more SBCs.

The SBC supports two types of transcoding:

- Transcoding After Rejection, page 40-3
- Codec Filtering, page 40-6
- Configuring Transcoding After Rejection, page 40-7
- Configuring Codec Filtering Transcoding, page 40-10

Transcoding After Rejection

The SBC automatically brings the transcoding device into use for any call requiring transcoding between these codecs, as long as the Call Admission Control (CAC) policy configuration does not preclude the transcoder service from being supplied for the call. When a call that requires transcoding is set up, the SBE goes through the following steps:

- Receives an initial signaling request from the calling endpoint. This triggers the SBC to perform initial call setup on the incoming and outgoing local media termination points. The SBC then forwards the set up request towards the callee.
- Receives a response from the called endpoint that indicates that none of the codecs in the initial request are acceptable. These responses include:
  - 415—Unsupported media type (SIP)
  - 488—Not acceptable here (SIP)
  - Failure to identify common codec during Terminal Capability Exchange procedure of H.245 protocol.
This triggers the SBC to bring a transcoder into the call that is inserted in the media path between the incoming and outgoing local media terminations. A new request is sent to the called endpoint, indicating the new codec type generated by the transcoder.

- SBE may then have to iterate through the list of codecs the transcoder supports until it finds one that is acceptable to the called endpoint. When this is done, the call is connected and media transmission begins.
Figure 40-1 shows where the transcoder sits in the network, and the path taken by the media in a transcoded call.

**Figure 40-1 Transcoding Configuration**

![Transcoding Configuration Diagram](image)

**Note**

Although Figure 40-1 shows two DBEs, transcoding is possible with a single DBE. With a single DBE, the media flows through the DBE twice, once on its way from the sending endpoint to the transcoder and a second time as it flows from the transcoder to the receiving endpoint.

For the Session Border Controller (SBC) to program the transcoder, it must be registered. The transcoding device acts as an H.248 media gateway, so it needs to be configured with the IP address and port of the SBE or SBC to connect to. The SBE or SBC acts as an H.248 Media Gateway Controller. See the documentation for your transcoder device for notes on how to do this. The documentation for the Cisco MGX 8880 Media Gateway can be found at:


In addition, the SBE must have the following specific configuration:

- An H.248 control address and port must be configured (using the `sbe control address ipv4` and `sbe control address h248 port` commands). By default, this is on port 2944, and it is the address and port to which the transcoder must connect.

- An explicit media gateway needs to be configured (using the `sbe media-gateway ipv4` command). The explicit media gateway must have its list of supported codecs defined so that the SBC knows which codecs the transcoder can translate between, and it must be identified as a transcoder (using the `sbe media-gateway ipv4 codecs` and `sbe media-gateway ipv4 transcoder` commands).

- The `show sbc sbe media-gateway-associations` command can be used to check that the transcoder has correctly registered with the SBE. If this has happened, the transcoder should appear in the list of known media gateways with an active association.

For configuration step information, see the “Configuring Transcoding After Rejection” section on page 40-7.
Troubleshooting Tip for Media-Timeout Transcoded Call Using a VXSM card

A Cisco MGX 8880 equipped with one or more Cisco Voice Switch Service Module (VXSM) card sets can operate as a media gateway. In a network where the SBC uses the Cisco MGX 8880 as a transcoding device to act as an H.248 media gateway, some additional configuration is required on the VXSM card for media-timeout to work properly in a transcoded call.

The following additional steps need to be configured on the VXSM card in the Cisco MGX 8880 Media Gateway:

---

**Step 1**
Enable the RTCP control with the following command.

```
InteropMGX.4.VXSM.a > cnfdspparam -control 1
```

**Step 2**
Set the RTCP timer control to startRtpOrRtcpPktRcvd with the following command:

```
InteropMGX.4.VXSM.a > cnfdspparam -rtcptm 3
```

**Step 3**
Verify that the settings are correct using the following command to show a list of DSP parameters:

```
InteropMGX.4.VXSM.a > dspdspparam
```

---

### List DSP Parameters

<table>
<thead>
<tr>
<th>SID Payload Type</th>
<th>decimal</th>
<th><strong>&lt;=== RTCP control enabled</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>RTCP Control</td>
<td>true</td>
<td></td>
</tr>
<tr>
<td>RTCP Interval(milliseconds)</td>
<td>5000</td>
<td></td>
</tr>
<tr>
<td>RTCP Interval Multiplier</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>VAD Adaptive</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>G.711 PLC</td>
<td>none</td>
<td></td>
</tr>
<tr>
<td>DTMF Power Level (0.1 dBm)</td>
<td>-120</td>
<td></td>
</tr>
<tr>
<td>DTMF Power Twist (0.1 dB)</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>RTCP Timer Control</td>
<td>startRtpOrRtcpPktRcvd</td>
<td><strong>&lt;=Timer Control properly set</strong></td>
</tr>
<tr>
<td>VQM Control</td>
<td>disable</td>
<td></td>
</tr>
<tr>
<td>RTPXR Control</td>
<td>enable</td>
<td></td>
</tr>
<tr>
<td>RTPXR Report Frequency</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>VQM Default Minimum Gap</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>RTPXR external R factor</td>
<td>127</td>
<td></td>
</tr>
<tr>
<td>SES Threshold (ms)</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>Voice IPIP mode</td>
<td>normal</td>
<td></td>
</tr>
</tbody>
</table>

---

For more information on the VXSM card, see the “VXSM as a Transcoding Gateway” chapter in the Cisco Voice Switch Service Module (VXSM) Configuration Guide Release 5.5 at:


---

**Codec Filtering**

The SBC allows you to restrict which codecs a particular call, caller and callee are allowed to use by whitelisting certain codecs. Initially all recognized codecs are on the whitelist. If a codec is requested which is absent from the call, caller, or callee codec whitelist, then the call still proceeds, but the forbidden codecs are removed from the offer and media gate configuration.
By supporting caller and callee codec lists, the SBC is able to make more intelligent transcoding decisions. If the codec support of either the calling or the called endpoint is known, then setting the caller and/or callee lists in a CAC policy is appropriate. However it may be that other considerations, such as the source adjacency, will affect the codec decision, in which case the per-call codec list can still be used.

For example, if the caller and callee codec lists are set to 'A and B', then all calls would use codec A. However, if a call had come across a transit network X (as indicated by the source adjacency) that only supported codec B, then the user could have an extra policy matching on source adjacency X, setting the per-call codec list to B. Calls crossing network X would then be forced to use codec B.

You can also limit the minimum packetization period of each codec, by configuring a value for the lowest acceptable minimum packetization period for each permitted codec. If a session is requested with a packetization period below this limit, the call still proceeds, but SBC increases the packetization period to the configured minimum value.

For configuration step information, see the ?Sparanum>Configuring Codec Filtering Transcoding? section on page 40-10.

**Configuring Transcoding After Rejection**

In this configuration area, the user supplies a configuration for a list of remote media gateways that may need to be managed by the SBE. This is not required when transcoding is not needed.

The primary reason for transcoding configurations is to configure the capabilities of external media transcoding devices when these devices cannot be discovered automatically. In-band auto-discovery of transcoder capabilities is currently not supported. Therefore, this step must be done when configuring all connections to all current remote transcoding devices.

Transcoding configurations can be skipped if the described reason does not apply.

By default, media gateways are allowed to connect whether or not configuration has been supplied for them. To help avoid configuration errors, the SBE logs a warning if an incoming connection is received from a media gateway that is not a DBE and does not have transcoding configured.

The basic steps for implementing transcoding are as follows:

1. Configure the IP address, port, and transport protocol for H.248 media gateway controller on SBC. This step may not be required if the Media Gateway Controller has already been configured.
2. Configure the media gateway IP address.
3. Configure the codecs to be transcoded (for example, between ITU-T G.711 ulaw and ITU-T G.729A).
4. Specify the media gateway as a transcoder.
5. Activate SBE.

This task implements transcoding for SBC.

Once configured, the SBC automatically brings the transcoding device into use for any call requiring transcoding between the codecs as long as the call admission control (CAC) policy configuration does not preclude the transcoder service from being supplied for the call using the `transcode-deny` command (See the Configuring Call Admission Control Policy Sets, CAC Tables, and Global CAC Policy Sets section in the ?Sparatext(CT_ChapTitle)?>? module).
### Note
In an H.323 adjacency configuration, you must use the `h245-tunnel disable` command for H.323 FastStart transcoded calls.

### SUMMARY STEPS

1. configure terminal
2. `sbc service-name`
3. `sbe`
4. `control address h248 index index-number`
5. `port port-number`
6. `ipv4 ipv4_IP_address`
7. `transport [transport-type]`
8. `exit`
9. `media-gateway ipv4 IPv4-IP-address`
10. `codecs codec-list`
11. `transcoder`
12. `exit`
13. `activate`
14. `end`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enables global configuration mode.</td>
</tr>
</tbody>
</table>
| **Example:**  
Router# configure terminal | |
| **Step 2** `sbc service-name` | Enters the mode of an SBC service. Use the `service-name` argument to define the name of the service. |
| **Example:**  
Router(config)# sbc mySbc | |
| **Step 3** `sbe` | Enters the mode of an SBE entity within a SBC service. |
| **Example:**  
Router(config-sbc)# sbe | |
| **Step 4** `control address h248 index index-number` | Configures an SBE to use a given IPv4 H.248 control address. |
| **Example:**  
Router(config-sbc-sbe)# control address h248 index 0 | |
### Command or Action | Purpose |
--- | --- |
**Step 5** | port port-number  
**Example:**  
Router(config-sbc-sbe-ctrl-h248)# port 2000 | Configures an SBE to use a given IPv4 H.248 port for H.248 communications. |
**Step 6** | ipv4 ipv4-IP-address  
**Example:**  
Router(config-sbc-sbe-ctrl-h248)# ipv4 1.1.1.1 | Configures an SBE to use a given IPv4 H.248 control address. |
**Step 7** | transport [transport-type]  
**Example:**  
Router(config-sbc-sbe-ctrl-h248)# transport udp | Configures transport type for H.248 communications. |
**Step 8** | exit  
**Example:**  
Router(config-sbc-sbe-ctrl-h248)# exit | Exits the current configuration mode. |
**Step 9** | media-gateway ipv4 IPv4-IP-address  
**Example:**  
Router(config-sbc-sbe)# media-gateway ipv4 10.0.0.1 | Configures a media gateway. |
**Step 10** | codecs codec-list  
**Example:**  
Router(config-sbc-sbe-mg)# codecs m=audio 1234 RTP/AVP 0 18,a=rtpmap:0 PCMU/8000,a=rtpmap:18 G729A/8000 | Configures the codecs supported by the media gateway. Enters into media gateway codecs configuration mode. |
**Step 11** | transcoder  
**Example:**  
Router(config-sbc-sbe-mg-codecs)# transcoder | Configures the media gateway with transcoder support. |
**Step 12** | exit  
**Example:**  
Router(config-sbc-sbe-mg-codecs)# exit | Exits media gateway codecs configuration mode to the sbe command mode level. |
**Step 13** | activate  
**Example:**  
Router(config-sbc-sbe-mg)# activate | Initiates the SBC service after all SBE address configuration has been successfully committed. |
**Step 14** | end  
**Example:**  
Router(config-sbc)# end | Ends the configuration session. |
Configuring Codec Filtering Transcoding

Configure codec filtering transcoding as shown below.

SUMMARY STEPS

1. configure terminal
2. sbc service-name
3. sbe
4. media-gateway ipv4 IPv4_IP_address
5. codecs codec-list
6. transcoder
7. exit
8. cac-policy-set
9. first-cac-table
10. cac-table
11. table-type policy-set
12. entry entry-num
13. caller-codec-list list-name
14. exit
15. exit
16. exit
17. codec-list list-name
18. codec codec-name
19. exit
20. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td></td>
</tr>
<tr>
<td>configure terminal</td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td></td>
</tr>
<tr>
<td>sbc service-name</td>
<td>Enters the mode of an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Use the service-name argument to define the name of the SBC service.</td>
</tr>
<tr>
<td>Router(config)# sbc mySBC</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><strong>sbe</strong> Enters the mode of an SBE entity within a SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc)# sbe</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td><strong>media-gateway ipv4 IPv4-IP-address</strong> Configures a media gateway.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe)# media-gateway ipv4 10.0.0.1</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td><strong>codecs codec-list</strong> Configures the codecs supported by the media gateway.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-mg)# codecs m=audio 1234 RTP/AVP 0 18,a=rtpmap:0 PCMU/8000,a=rtpmap:18 G729A/8000</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td><strong>transcoder</strong> Configures the media gateway with transcoder support.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-mg-codecs)# transcoder</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td><strong>exit</strong> Exits the media gateway configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-mg-codecs)# exit</td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td><strong>cac-policy-set</strong> Enters the CAC policy submode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe)# cac-policy-set 1</td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td><strong>first-cac-table</strong> Creates or configures the first admission control table.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-cacpolicy)# first-cac-table 1</td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td><strong>cac-table</strong> Creates or configures an admission control table.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-cacpolicy)# cac-table 1</td>
</tr>
<tr>
<td><strong>Step 11</strong></td>
<td><strong>table-type policy-set</strong> Configures the Policy Set table type of Call Admission Control (CAC) table.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-cacpolicy-cactable)# table-type policy-set</td>
</tr>
<tr>
<td><strong>Step 12</strong></td>
<td><strong>entry</strong> Creates or modifies an entry in a table.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-cacpolicy-cactable)# entry 1</td>
</tr>
</tbody>
</table>
### Information About Transcoding

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td><code>caller-codec-list list-name</code></td>
<td>Lists the codecs which the caller leg of a call is allowed to use.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Router(config-sbc-sbe-cacpolicy-cactable-entry) # caller-codec-list my_codecs</code></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td><code>exit</code></td>
<td>Exits the CAC table entry configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Router(config-sbc-sbe-cacpolicy-cactable-entry) # exit</code></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td><code>exit</code></td>
<td>Exits the CAC table configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Router(config-sbc-sbe-cacpolicy-cactable)# exit</code></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td><code>exit</code></td>
<td>Exits the CAC policy configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Router(config-sbc-sbe-cacpolicy)# exit</code></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td><code>codec-list list-name</code></td>
<td>Creates a codec list and enters the Codec list configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Router(config-sbc-sbe)# codec-list my_codecs</code></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td><code>codec codec-name</code></td>
<td>Adds a codec to a codec list.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Router(config-sbc-sbe-codec-list)# codec PCMU</code></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td><code>exit</code></td>
<td>Exits the Codec list configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Router(config-sbc-sbe-codec-list)# exit</code></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td><code>end</code></td>
<td>Ends the configuration session.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Router(config-sbc-sbe)# end</code></td>
<td></td>
</tr>
</tbody>
</table>
The example below is a configuration of transcoding after rejection.

Router(config)# sbc mySbc
Router(config-sbc)# sbe
Router(config-sbc-sbe)# control address h248 index 1
Router(config-sbc-sbe-ctrl-h248)# port 2000
Router(config-sbc-sbe-ctrl-h248)# ipv4 88.88.133.2
Router(config-sbc-sbe-ctrl-h248)# transport udp
Router(config-sbc-sbe)# media-gateway ipv4 10.0.0.1
Router(config-sbc-sbe-mg)# codecs m=audio 1234 RTP/AVP 0 18,a=rtpmap:0 PCMU/8000,a=rtpmap:18 G729A/8000
Router(config-sbc-sbe-mg-codecs)# transcoder
Router(config-sbc-sbe-mg-codecs)# exit
Router(config-sbc-sbe-mg)# activate
Router(config-sbc)# end

Below is an example of codec filtering transcoding.

Router(config)# ip route 10.0.20.33 255.255.255.255 10.130.10.33
Router(config)# ip route 0.0.0.0 0.0.0.0 10.74.50.114
Router(config)# ip route 0.0.0.0 0.0.0.0 10.130.10.1
Router(config)# snmp-server community cisco group Network-Monitor
Router(config)# snmp-server community public group Network-Monitor
Router(config)# snmp-server community private group Network-Monitor

Router# configure terminal
Router(config)# sbc sbc-11
Router(config-sbc)# sbe
Router(config-sbc-sbe)# media-gateway ipv4 10.100.181.2
Router(config-sbc-sbe-mg)# codecs m=audio 20000 RTP/AVP 0 8 18,a=rtpmap:0 PCMU/8000,a=rtpmap:18 G729/8000
Router(config-sbc-sbe-mg)# transcoder

Router(config-sbc-sbe)# control address h248 index 1
Router(config-sbc-sbe-ctrl-h248)# ipv4 10.130.10.4
Router(config-sbc-sbe-ctrl-h248)# transport udp

Router(config-sbc-sbe)# adjacency sip SIPP81
Router(config-sbc-sbe-adj-sip)# nat force-off
Router(config-sbc-sbe-adj-sip)# preferred-transport udp
Router(config-sbc-sbe-adj-sip)# redirect-mode pass-through
Router(config-sbc-sbe-adj-sip)# authentication nonce timeout 300
Router(config-sbc-sbe-adj-sip)# signaling-address ipv4 10.130.10.4
Router(config-sbc-sbe-adj-sip)# signaling-port 5060
Router(config-sbc-sbe-adj-sip)# remote-address ipv4 10.0.244.81 255.255.255.255
Router(config-sbc-sbe-adj-sip)# signaling-peer 10.0.244.81
Router(config-sbc-sbe-adj-sip)# signaling-peer-port 5060
Router(config-sbc-sbe-adj-sip)# dbe-location-id 0
Router(config-sbc-sbe-adj-sip)# reg-min-expiry 3000
Router(config-sbc-sbe-adj-sip)# attach

Router(config-sbc-sbe)# adjacency sip SIPP91
Router(config-sbc-sbe-adj-sip)# nat force-off
Router(config-sbc-sbe-adj-sip)# preferred-transport udp
Router(config-sbc-sbe-adj-sip)# redirect-mode pass-through
Router(config-sbc-sbe-adj-sip)# authentication nonce timeout 300
Router(config-sbc-sbe-adj-sip)# signaling-address ipv4 10.130.10.4
Configuration Examples for Implementing Transcoding

Router(config-sbc-sbe-adj-sip)# signaling-port 5060
Router(config-sbc-sbe-adj-sip)# remote-address ipv4 10.0.244.91 255.255.255.255
Router(config-sbc-sbe-adj-sip)# signaling-peer 10.0.244.91
Router(config-sbc-sbe-adj-sip)# signaling-peer-port 5060
Router(config-sbc-sbe-adj-sip)# dbe-location-id 0
Router(config-sbc-sbe-adj-sip)# reg-min-expiry 3000
Router(config-sbc-sbe-adj-sip)# attach

Router(config-sbc-sbe)# sip inherit profile preset-core
Router(config-sbc-sbe)# cac-policy-set 1
Router(config-sbc-sbe-cacpolicy)# first-cac-table table
Router(config-sbc-sbe-cacpolicy)# first-cac-scope call
Router(config-sbc-sbe-cacpolicy)# averaging-period 60
Router(config-sbc-sbe)# cac-table table
Router(config-sbc-sbe-cacpolicy-cactable)# match-type adjacency
Router(config-sbc-sbe-cacpolicy-cactable)# entry 1
Router(config-sbc-sbe-cacpolicy-cactable-entry)# match-value SIPP81
Router(config-sbc-sbe-cacpolicy-cactable-entry)# action cac-complete
Router(config-sbc-sbe-cacpolicy-cactable-entry)# max-bandwidth 64009 Gbps
Router(config-sbc-sbe-cacpolicy-cactable-entry)# max-updates 4294967295
Router(config-sbc-sbe-cacpolicy-cactable-entry)# max-channels 4294967295
Router(config-sbc-sbe-cacpolicy-cactable-entry)# early-media-type full-duplex
Router(config-sbc-sbe-cacpolicy-cactable-entry)# early-media-timeout 0
Router(config-sbc-sbe-cacpolicy-cactable-entry)# caller-codec-list allow711u
Router(config-sbc-sbe-cacpolicy-cactable-entry)# callee-privacy never
Router(config-sbc-sbe-cacpolicy-cactable-entry)# callee-hold-setting standard
Router(config-sbc-sbe-cacpolicy-cactable-entry)# complete

Router (config-sbc-sbe)# active-cac-policy-set 1

Router (config-sbc-sbe)# retry-limit 3

Router (config-sbc-sbe)# call-policy-set 1
Router(config-sbc-sbe-rtgpolicy)# first-call-routing-table table
Router(config-sbc-sbe-rtgpolicy-rtgtable)# entry 1
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# dst-adjacency SIPP91
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# match-address 318X

Router (config-sbc-sbe)# active-call-policy-set 1

Router(config-sbc-sbe)# sip max-connections 2
Router(config-sbc-sbe)# sip timer
Router(config-sbc-sbe-sip-tmr)# tcp-idle-timeout 120000
Router(config-sbc-sbe-sip-tmr)# tls-idle-timeout 3600000
Router(config-sbc-sbe-sip-tmr)# udp-response-linger-period 32000
Verification

Use the following `show sbc sbe media-gateway-associations` command to display a list of known media gateways with an active association and to verify the operation:

The following example shows the SBC and media communications.

```bash
Router# show sbc slt-n2 sbe media-gateway-associations
SBC Service "slt-n2"
  Media gateway 192.169.125.1:2944
    Gateway Protocol = megaco
    Transport Protocol = UDP
```

```bash
Router(config-sbc-sbe)## udp-first-retransmit-interval 500
Router(config-sbc-sbe)## udp-max-retransmit-interval 4000
Router(config-sbc-sbe)## invite-timeout 180

Router (config-sbc-sbe)# codec-list allow711u
Router(config-sbc-sbe-codec-list)# codec PCMU

Router (config-sbc-sbe)# codec-list allowg729
Router(config-sbc-sbe-codec-list)# codec G729

Router(config-sbc-sbe)# h323
Router (config-sbc-sbe-h323)# ras timeout arq 5000
Router (config-sbc-sbe-h323)# ras retry arq 2
Router (config-sbc-sbe-h323)# ras timeout brq 3000
Router (config-sbc-sbe-h323)# ras retry brq 2
Router (config-sbc-sbe-h323)# ras timeout drq 3000
Router (config-sbc-sbe-h323)# ras retry drq 2
Router (config-sbc-sbe-h323)# ras timeout grq 5000
Router (config-sbc-sbe-h323)# ras retry grq 2
Router (config-sbc-sbe-h323)# ras timeout rrq 3000
Router (config-sbc-sbe-h323)# ras retry rrq 2
Router (config-sbc-sbe-h323)# ras rrq ttl 60
Router (config-sbc-sbe-h323)# ras timeout urq 3000
Router (config-sbc-sbe-h323)# ras retry urq 1
Router (config-sbc-sbe-h323)# h225 timeout proceeding 10000
Router (config-sbc-sbe-h323)# h225 timeout establishment 180000
Router (config-sbc-sbe-h323)# h225 timeout setup 4000
Router (config-sbc-sbe-h323)# ras rrq keeplive 45000

Router(config-sbc-sbe)# h323
Router (config-sbc-sbe-adj-h323)# adjacency timeout 30000

Router(config-sbc-sbe)# blacklist
Router(config-sbc-sbe-blacklist)# global

Router(config-sbc-sbe)# blacklist
Router(config-sbc-sbe-blacklist)# address-default

Router(config-sbc-sbe)# redirect-limit 2
Router(config-sbc-sbe)# deact-mode normal
Router (config-sbc-sbe)# activate

Router(config-sbc)# dbe
Router(config-sbc-dbe)# location-id 0
Router(config-sbc-dbe)# media-timeout 360
Router(config-sbc-dbe)# deact-mode normal
Router (config-sbc-dbe)# activate
```
Voice Transcoding Per Adjacency Statistics

The Voice Transcoding Per Adjacency Statistics feature provides the transcoding statistics to the user for voice calls at both global and adjacency levels. The feature analyzes the consumption of the cards, such as the DSP cards, that provide the transcoding functions.

The transcoding statistics include the following information:

- The number of active transcoding media stream for each codec pair over several summary periods at global and adjacency scopes. The statistic also provides a high water mark for the corresponding codec pair.
- The number of active transcoding calls both per-adjacency and globally are listed. The statistics can be listed both at global and adjacency scopes for the list of codec pairs.
- The statistics display the codec names for the following codecs, if the transcoding call uses any other codecs, the codec name is displayed as Other:
  - G711A
  - G711U
  - G729
  - GSM
  - T38
  - CLEAR

Configuring the Voice Transcoding Per Adjacency Statistics

This task shows how to configure the Voice Transcoding Per Adjacency Statistics feature, list the transcoding statistics as per the scope and summary period, and also reset the transcoding statistics.

SUMMARY STEPS

1. configure terminal
2. sbc sbc-name
3. sbe
4. transcoding-stats enable
5. end
6. show sbc sbc-name sbe transcoding-stats {global | adjacency adjacency-name} {current15mins | current5mins | currentday | currenthour | current-indefinite | previous15mins | previous5mins | previousday | previoushour}
7. clear sbc sbc-name sbe transcoding-stats [global | adjacency adjacency-name] [all | current-indefinite]
# DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td>Example: Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc sbc-name</td>
<td>Enters the SBC service mode. Use the <em>sbc-name</em> argument to define the name of the service.</td>
</tr>
<tr>
<td>Example: Router(config)# sbc mySBC</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Enters the SBE entity mode within a SBC service.</td>
</tr>
<tr>
<td>Example: Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> transcoding-stats enable</td>
<td>Enables or disables the transcoding related statistics for the SBC.</td>
</tr>
<tr>
<td>Example: Router(config-sbc-sbe)# no transcoding-stats enable</td>
<td>The following warning is issued and the user needs to confirm <em>y</em> (Yes) or <em>n</em> (No) to enable or disable the transcoding statistics:</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> This will re-activate the SBC, and existing calls will be impacted[confirm]</td>
</tr>
<tr>
<td></td>
<td>By default, the transcoding related statistics for the SBC is enabled.</td>
</tr>
<tr>
<td><strong>Step 5</strong> end</td>
<td>Exits the SBE entity mode and enters the Privileged Exec mode.</td>
</tr>
<tr>
<td>Example: Router(config-sbc-sbe)# end</td>
<td></td>
</tr>
</tbody>
</table>
### Voice Transcoding Per Adjacency Statistics

#### Command or Action

| Step 6 | show sbc sbc-name sbe transcoding-stats {global | adjacency adjacency-name} {current15mins | current5mins | currentday | currenthour | current-indefinite | previous15mins | previous5mins | previousday | previoushour} |
|---|---|
| **Example:** | Router# show sbc mySBC sbe transcoding-stats adjacency SIPP current5mins |

#### Purpose
- Lists the voice transcoding statistics for the required scope and summary period.
  - **adjacency**—Lists the transcoding statistics for the specified adjacency.
  - **global**—Lists the globally scoped statistics for the entire SBC.
  - **current15mins**—Lists the statistics for the current 15 minute interval.
  - **current5mins**—Lists the statistics for the current 5 min. interval.
  - **currentday**—Lists the statistics for the current day from midnight.
  - **currenthour**—Lists the statistics for the current hour.
  - **currentindefinite**—Lists the statistics for the period since the last explicit reset.
  - **previous15mins**—Lists the statistics for the previous 15 minute interval.
  - **previous5mins**—Lists the statistics for the previous 5 min. interval.
  - **previousday**—Lists the statistics for the previous day.
  - **previoushour**—Lists the statistics for the previous hour.

#### Step 7

| clear sbc sbc-name sbe transcoding-stats {global | adjacency adjacency-name} {all | current-indefinite} |
|---|---|
| **Example:** | Router# clear sbc mySBC sbe adjacency SIPP all |

Clears the transcoding statistics for all or current-indefinite summary period.
- **adjacency**—Clears statistics for the adjacency.
- **global**—Clears the global transcoding statistics.
- **all**—Clears statistics for all summary periods.
- **currentindefinite**—Clears statistics for only the current-indefinite period.

The following example shows the output of the `show sbc sbe transcoding-stats adjacency current15mins` command:

```
Router# show sbc mySBC sbe transcoding-stats adjacency SIPP current15mins

<table>
<thead>
<tr>
<th>Codec1</th>
<th>Codec2</th>
<th>Transcoded Stream</th>
<th>HWM of TranscodedStream</th>
<th>Last Reset</th>
</tr>
</thead>
<tbody>
<tr>
<td>G711A</td>
<td>G711U</td>
<td>4</td>
<td>10</td>
<td>2010/09/10 19:27:15</td>
</tr>
</tbody>
</table>
```
Media Gateway-Assisted DTMF Interworking

The SBC enables inband DTMF interworking using media gateway switches such as the MGX 8880. A Cisco MGX 8880 is used with DTMF interworking in the following scenarios:

- As a transcoder—DTMF interworking between media plane and signaling is supported.
- As an inband DTMF extractor or injector.

The SBC supports two types of media plane DTMF:

- RFC2833 (telephone-event)
- Inband DTMF—DTMF inband audio stream, such as G.711. To support inband DTMF, MGX performs the following tasks:
  - Monitors the audio stream.
  - Extracts the DTMF signal.
  - Reports or injects the DTMF signal into the voice band, and vice versa.

DTMF Interworking with MGX as Transcoder

When the SBC uses an external transcoder, such as MGX, DTMF interworking is supported for the following:

- Between media and signaling in a call.
- For both, negotiated transcoding and transcoding provisioned through the use of codec lists.
- Between supported media formats, such as RFC2833, and supported SIP signaling formats, such as INFO or NOTIFY.

Inband DTMF Support—Interworking Without a Transcoder

The SBC supports a call or adjacency policy to indicate when an inband DTMF tone is monitored. Monitoring an inband DTMF tone can either be forced, or an optional task in the absence of any other DTMF support.

The SBC enables interworking between any two of the three supported DTMF formats, which include:

- Inband
- RFC 2833 telephone events
- Signaling.

In the event of a failover, active calls using any DTMF interworking option are protected, and the interworking capability is retained on restoration.

The SBC provides a per-adjacency option to enforce an inband DTMF-compatible codec negotiation if no other methods are available for receiving or sending DTMF.
Chapter 40  Implementing Transcoding

Configuring Inband DTMF Interworking

To configure inband DTMF interworking, perform the following steps.

Note
The caller and callee commands have been used in this procedure. In some scenarios, the branch command can be used as an alternative to the caller and callee command pair. The branch command has been introduced in Release 3.5.0. See the Configuring Directed Nonlimiting CAC Policies section on page 7-37 for information about this command.

SUMMARY STEPS

1. configure terminal
2. sbc service-name
3. sbe
4. cac-policy-set policy-set-id
5. first-cac-scope scope-name
6. first-cac-table table-name
7. cac-table table-name
8. table-type {policy-set | limit {list of limit tables}}
9. entry entry-id
10. cac-scope {list of scope options}
11. callee inband-dtmf-mode {always | inherit | maybe | never}
12. caller inband-dtmf-mode {always | inherit | maybe | never}
13. complete
14. active-call-policy-set policy-set-id
15. end
16. show sbc service-name sbe cac-policy-set id table name entry entry

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure terminal</td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 2 sbc service-name</td>
<td>Enters the submode of an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# sbc mysbc</td>
<td></td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
</tr>
<tr>
<td>------</td>
<td>------------------</td>
</tr>
<tr>
<td>3</td>
<td><code>sbe</code></td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td><code>Router(config-sbc)# sbe</code></td>
</tr>
<tr>
<td>4</td>
<td><code>cac-policy-set policy-set-id</code></td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td><code>Router(config-sbc-sbe)# cac-policy-set 1</code></td>
</tr>
<tr>
<td>5</td>
<td><code>first-cac-scope scope-name</code></td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
</tr>
<tr>
<td></td>
<td><em>adj-group</em></td>
</tr>
<tr>
<td></td>
<td><em>call</em></td>
</tr>
<tr>
<td></td>
<td><em>category</em></td>
</tr>
<tr>
<td></td>
<td><em>dst-account</em></td>
</tr>
<tr>
<td></td>
<td><em>dst-adj-group</em></td>
</tr>
<tr>
<td></td>
<td><em>dst-adjacency</em></td>
</tr>
<tr>
<td></td>
<td><em>dst-number</em></td>
</tr>
<tr>
<td></td>
<td><em>global</em></td>
</tr>
<tr>
<td></td>
<td><em>src-account</em></td>
</tr>
<tr>
<td></td>
<td><em>src-adj-group</em></td>
</tr>
<tr>
<td></td>
<td><em>src-adjacency</em></td>
</tr>
<tr>
<td></td>
<td><em>src-number</em></td>
</tr>
<tr>
<td></td>
<td>Features can be enabled or disabled per adjacency group through CAC configuration the same way this is done per individual adjacency.</td>
</tr>
<tr>
<td>6</td>
<td><code>first-cac-table table-name</code></td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td><code>Router(config-sbc-sbe-cacpolicy)# first-cac-table first_policy_table</code></td>
</tr>
</tbody>
</table>
### Step 7

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>cac-table table-name</code></td>
<td>Enters the mode for configuration of an admission control table (creating one, if necessary) within the context of an SBE policy set.</td>
</tr>
</tbody>
</table>

**Example:**

Router(config-sbc-sbe-cacpolicy)# cac-table first_policy_table
Command or Action

| Step 8 | table-type (policy-set | limit (list of limit tables)) |
|--------|------------------------|

Example:
Router(config-sbc-sbe-cacpolicy-cactable)#
table-type policy-set

Purpose

Configures the table type of a CAC table within the context of an SBE policy set. The list of limit tables argument controls the syntax of the match-value fields of the entries in the table. Possible list of limit tables values are:

- **account**—Compare with name of the account.
- **adj-group**—Compare with name of the adjacency group.
- **adjacency**—Compare with name of the adjacency.
- **all**—No comparison type. All events match this type.
- **call-priority**—Compare with call priority.
- **category**—Compare with number analysis-assigned category.
- **dst-account**—Compare with name of the destination account.
- **dst-adj-group**—Compare with name of the destination adjacency group.
- **dst-adjacency**—Compare with name of the destination adjacency.
- **dst-prefix**—Compare with beginning of the dialed digit string.
- **event-type**—Compare with CAC policy event types.
- **src-account**—Compare with name of the source account.
- **src-adj-group**—Compare with name of the source adjacency group.
- **src-adjacency**—Compare with name of the source adjacency.
- **src-prefix**—Compare with beginning of the calling number string.

Note For Limit tables, the event, message, call matches only a single entry.

Features can be enabled or disabled per adjacency group through CAC configuration the same way this is done per individual adjacencies. The adj-group table type matches with either the source or the destination adjacency group.

When the **policy-set** keyword is specified, use the **cac-scope** command to configure the scope within each entry at which limits are applied in a CAC Policy Set table.

Note For Policy Set tables, the event, call or message, is applied to all entries in this table.
### Media Gateway-Assisted DTMF Interworking

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 9**

**entry entry-id**

**Example:**
Router(config-sbc-sbe-cacpolicy-cactable)# entry 1

Enters the mode to create or modify an entry in an admission control table.

| **Step 10**

**cac-scope** *(list of scope options)*

**Example:**
Router(config-sbc-sbe-cacpolicy-cactable-entry)# cac-scope call

Configures the scope within each entry at which limits are applied in a Policy Set table.

Only per-call scope can be configured when using the *codec-restrict-to-list* command.

**list of scope options**—Specifies one of the following strings used to match events:

- **account**—Events that are from the same account.
- **adjacency**—Events that are from the same adjacency.
- **adj-group**—Events that are from members of the same adjacency group.
- **call**—Scope limits are per single call.
- **category**—Events that have the same category.
- **dst-account**—Events that are sent to the same account.
- **dst-adj-group**—Events that are sent to the same adjacency group.
- **dst-adjacency**—Events that are sent to the same adjacency.
- **dst-number**—Events that have same destination.
- **global**—Scope limits are global.
- **src-account**—Events that are from the same account.
- **src-adj-group**—Events that are from the same adjacency group.
- **src-adjacency**—Events that are from the same adjacency.
- **src-number**—Events that have the same source number.
- **sub-category**—The limits specified in this scope apply to all the events sent to or received from members of the same subscriber category.
- **sub-category-pfx**—The limits specified in this scope apply to all events sent to or received from members of the same subscriber category prefix.
- **subscriber**—The limits specified in this scope apply to all the events sent to or received from individual subscribers

A device that is registered with a Registrar server.
### Command or Action

| Step 11 | callee inband-dtmf-mode {always | inherit | maybe | never} |
|---------|--------------------------------------------------|

**Example:**
```
Router(config-sbc-sbe-cacpolicy-cactable-entry)
# callee inband-dtmf-mode always
```

To configure a DTMF inband mode for the callee side, use the `callee inband-dtmf-mode` command in the CAC table configuration mode. To deconfigure the DTMF inband mode for the callee side, use the `no` form of this command.

The `callee inband-dtmf-mode` specifies one of the following strings:

- **always**—The inband DTMF tones are always in use by an endpoint.
- **inherit**—The inband DTMF mode for an endpoint is not affected by the CAC entry.
- **maybe**—The inband DTMF tones are used by an endpoint unless signaling indicates that an alternative format for a DTMF is in use.
- **never**—An endpoint never uses inband DTMF.

| Step 12 | caller inband-dtmf-mode {always | inherit | maybe | never} |
|---------|--------------------------------------------------|

**Example:**
```
Router(config-sbc-sbe-cacpolicy-cactable-entry)
# caller inband-dtmf-mode never
```

To configure the DTMF inband mode for the caller side, use the `caller inband-dtmf-mode` command in the CAC table configuration mode. To deconfigure the DTMF inband mode for the caller side, use the no form of this command.

The `caller inband-dtmf-mode` specifies one of the following strings:

- **always**—The inband DTMF tones are always in use by an endpoint.
- **inherit**—The inband DTMF mode for an endpoint is not affected by the CAC entry.
- **maybe**—The inband DTMF tones are used by an endpoint unless signaling indicates that an alternative format for a DTMF is in use.
- **never**—An endpoint never uses inband DTMF.

<table>
<thead>
<tr>
<th>Step 13</th>
<th>complete</th>
</tr>
</thead>
</table>

**Example:**
```
Router(config-sbc-sbe-cacpolicy)# complete
```

Completes the CAC-policy.

<table>
<thead>
<tr>
<th>Step 14</th>
<th>active-call-policy-set policy-set-id</th>
</tr>
</thead>
</table>

**Example:**
```
Router(config-sbc-sbe-cacpolicy)#
cac-policy-set global 1
```

Sets the active routing policy set within an SBE entity.
### Configuring Codecs to Support Inband DTMF

To configure the codecs to support inband DTMF, perform the following tasks:

#### SUMMARY STEPS

1. `configure terminal`
2. `sbc service-name`
3. `sbe`
4. `codec system system-name id`
5. `options {none | transrate | transcode | inband-dtmf}`
6. `end`

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td><code>configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Enters the submode of an SBC service.</td>
</tr>
<tr>
<td><code>sbc service-name</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Configures the submode of the SBE entity within an SBC service.</td>
</tr>
<tr>
<td><code>sbe</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Specifies the name of the system, analog-to-digital codec (enCOder/DECoder), and enters the Codec definition mode.</td>
</tr>
<tr>
<td><code>codec system system-id</code></td>
<td></td>
</tr>
</tbody>
</table>
Blended Transcoding

The Blended Transcoding feature enables the SBC to establish sessions without transcoding. Do not enable the Blended Transcoding feature in the following situations:

- When using H.323 or SIP-H.323 interworking calls
- When the calls are under transcoding video streams
- When the calls are in fax transcoding

The Blended Transcoding feature does not work with the following features:

- Media Bypass
- H.323 Calls and SIP-H.323 Interworking
- Late-Early Interworking
- Downstream Forking with Codec Change
- Local Call Transfer
- IMS (Gq and Rx)

Before you enable the Blended Transcoding feature, make sure that the DSP farm codec is already configured. For more information about the DSP farm codec configuration, see the Cisco Unified Border Element (SP Edition) Configuration Guide: Unified Model at:


Enabling Blended Transcoding

To enable the Blended Transcoding feature, perform the following steps:

SUMMARY STEPS

1. configure terminal
2. sbc sbc-name
3. sbe

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| 5    | options (none | transrate | transcode | inband-dtmf) | Configures the codec that will support voice inband DTMF. The values for the options are:  
- none  
- transrate  
- transcode  
- inband-dtmf |
|      | **Example:**      |         |
|      | Router(config-sbc-sbe-codec-def)# options inband-dtmf |         |
| 6    | end               | Exits the cac-policy-set configuration mode and enters the Privileged EXEC mode. |
|      | **Example:**      |         |
|      | Router(config-sbc-sbe-cacpolicy)# end |         |
**Blended Transcoding**

4. `codec list list-name`
5. `codec codec-name`
6. `cac-policy-set cac-policy-name`
7. `first-cac-table table-name`
8. `cac-table cac-table-name`
9. `table-type limit adjacency`
10. `entry entry-id`
11. `match-value string-value`
12. `blended-transcode`
13. `blended-codec-list codec-list-name`
14. `action cac-complete`
15. `complete`
16. `cac-policy-set global policy-set-id`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td></td>
</tr>
<tr>
<td><code>configure terminal</code></td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td></td>
</tr>
<tr>
<td><code>sbc sbc-name</code></td>
<td>Enables the SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config)# sbc mysbc</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td></td>
</tr>
<tr>
<td><code>sbe</code></td>
<td>Enters the mode of the SBE entity within an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-sbc)# sbe</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td></td>
</tr>
<tr>
<td><code>codec list list-name</code></td>
<td>Creates a codec list.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-sbc-sbe)# codec list my_codecs</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td></td>
</tr>
<tr>
<td><code>codec codec-name</code></td>
<td>Adds a codec to the codec list.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-sbc-sbe-codec-list)# codec PCMU</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td></td>
</tr>
<tr>
<td><code>cac-policy-set policy-set-id</code></td>
<td>Creates a new call admission control (CAC) policy set.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-sbc-sbe)# cac-policy-set 1</code></td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 7</td>
<td><code>first-cac-table table-name</code></td>
<td>Specifies the admission control table that should be processed first.</td>
</tr>
<tr>
<td>Example</td>
<td><code>Router(config-sbc-sbe-cacpolicy)#first-cac-table BlendedTranscodeTable</code></td>
<td></td>
</tr>
<tr>
<td>Step 8</td>
<td><code>cac-table cac-table-name</code></td>
<td>Creates an admission control table.</td>
</tr>
<tr>
<td>Example</td>
<td><code>Router(config-sbc-sbe-cacpolicy)# cac-table BlendedTranscodeTable</code></td>
<td></td>
</tr>
<tr>
<td>Step 9</td>
<td><code>table-type limit adjacency</code></td>
<td>Configures a CAC table type that determines the priority of the call to be used as a criterion in the CAC policy.</td>
</tr>
<tr>
<td>Example</td>
<td><code>Router(config-sbc-sbe-cacpolicy-cactable)# table-type limit adjacency</code></td>
<td></td>
</tr>
<tr>
<td>Step 10</td>
<td><code>entry entry-id</code></td>
<td>Creates an entry in the CAC table.</td>
</tr>
<tr>
<td>Example</td>
<td><code>Router(config-sbc-sbe-cacpolicy-cactable)# entry 1</code></td>
<td></td>
</tr>
<tr>
<td>Step 11</td>
<td><code>match-value string-value</code></td>
<td>Specifies the adjacency that is to be enabled with the Blended Transcoding feature.</td>
</tr>
<tr>
<td>Example</td>
<td><code>Router(config-sbc-sbe-cacpolicy-cactable-entry)# match-value SIP3</code></td>
<td></td>
</tr>
<tr>
<td>Step 12</td>
<td><code>blended-transcode</code></td>
<td>Enables the Blended Transcoding feature.</td>
</tr>
<tr>
<td>Example</td>
<td><code>Router(config-sbc-sbe-cacpolicy-cactable-entry)# blended-transcode</code></td>
<td></td>
</tr>
<tr>
<td>Step 13</td>
<td><code>blended-codec-list codec-list name</code></td>
<td>Configures a blended codec list.</td>
</tr>
<tr>
<td>Example</td>
<td><code>Router(config-sbc-sbe-cacpolicy-cactable-entry)# blended-codec-list my_codecs</code></td>
<td></td>
</tr>
<tr>
<td>Step 14</td>
<td><code>action cac-complete</code></td>
<td>Configures the action to be performed after the CAC entry in an admission control table; indicates that this CAC policy is complete.</td>
</tr>
<tr>
<td>Example</td>
<td><code>Router(config-sbc-sbe-cacpolicy-cactable-entry)# action cac-complete</code></td>
<td></td>
</tr>
</tbody>
</table>
The following example shows how to enable the Blended Transcoding feature:

codec list codec-a
  codec PCMU
  codec G729

cac-policy-set 1
  first-cac-table BlendedTranscodingTable
  cac-table BlendedTranscodingTable
  table-type limit adjacency
  entry 1
    match-value SIP3
    blended-transcode
    blended-codec-list codec-a
    action cac-complete
  complete

cac-policy-set global 1

---

Step 15

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>complete</td>
<td>Ends the configuration of the CAC table, and goes back to the SBC SBE configuration mode.</td>
</tr>
</tbody>
</table>

Example:

```
Router(config-sbc-sbe-cacpolicy-cactable-entry)
# complete
```

Step 16

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>cac-policy-set global policy-set-id</td>
<td>Activates the global CAC policy set within the SBE entity.</td>
</tr>
</tbody>
</table>

Example:

```
Router (config-sbc-sbe)# cac-policy-set global 1
```

---

**Configuration Example for Blended Transcoding**

The following example shows how to enable the Blended Transcoding feature:

codec list codec-a
  codec PCMU
  codec G729

cac-policy-set 1
  first-cac-table BlendedTranscodingTable
  cac-table BlendedTranscodingTable
  table-type limit adjacency
  entry 1
    match-value SIP3
    blended-transcode
    blended-codec-list codec-a
    action cac-complete
  complete

cac-policy-set global 1
Cisco Unified Border Element (SP Edition)—SPA DSP Services

The shared port adapter (SPA) digital signal processor (DSP) is a single-width, half-height, high-power, SPA module that can be used across multiple Cisco platforms. The SPA DSP is designed for DSP-based voice and video solutions in the SPAs on the Cisco mid-range and high-end routers.

In Cisco IOS XE Release 3.2S, the following SPA DSP features have been deployed on the Cisco ASR 1000 Series Router for the session border controller (SBC):

- Associating SBC configuration with a DSP farm profile.
- Voice transcoding and transrating support using onboard DSP services.
- Dual tone multifrequency (DTMF) interworking using onboard DSP services.
- VoIPv4 and VoIPv6 transcoding and transrating support.
- Transcoding, transrating, and DTMF interworking call control and signaling control.

Cisco Unified Border Element (SP Edition) was earlier known as Integrated Session Border Controller, and is referred to as SBC in this document.


For information about all the Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the Cisco IOS master commands list.

Feature History of SPA DSP on the Cisco Unified Border Element (SP Edition)

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 3.2S</td>
<td>The SPA DSP onboard services were introduced on the Cisco ASR 1000 Series Routers.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 3.3S</td>
<td>The Call Recovery feature was added.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 3.8S</td>
<td>The AMR-WB feature was supported on the SBC on the Cisco ASR 1000 Aggregation Services Routers.</td>
</tr>
</tbody>
</table>
Restrictions

The following restrictions are applicable to a SPA DSP:

- Voice, audio, and video conferencing are not supported.
- HA, system-level In-Service Software Upgrade (ISSU), and Nonstop Forwarding (NSF) are not supported.
- Video codecs are not supported.
- Although Online Insertion and Removal (OIR) is supported, the sessions going through a SPA at the time of removal are lost.
- The Cisco Unified Communications Manager is not supported.

Prerequisites for the SPA DSP Services

The DSP farm definition and SBC configuration and activation must be completed before transcoding the SBC calls. For more information about SPA configuration, see the “Configuring the Cisco DSP SPA for the ASR 1000 Series” chapter in Cisco ASR 1000 Series Aggregation Services Routers SIP and SPA Software Configuration Guide at:


Information About the SPA DSP Services

A SPA DSP contains digital signal processors and related hardware to provide voice transcoding capability for the SBC. In addition, Cisco Unified Border Element, Enterprise can use a SPA DSP for simple voice transcoding services.

You can find more information on terminating and generating the RTCP by the SPA-DSP at:
Transcoding the SBC

SBC transcoding is used for codec translation between two VoIP networks as part of the Data Border Element (DBE) functions. Figure 41-1 shows how a SPA DSP performs codec transcoding for unified SBC and Figure 41-2 shows how a SPA DSP performs codec transcoding for distributed SBC.

**Figure 41-1  SPA DSP Transcoding for Unified SBC**

**Figure 41-2  SPA DSP Transcoding for Distributed SBC**

The SPA DSP allows the translation of one type of media stream or codec to another type of media stream that uses different media encoding and decoding technologies. Other translation activities include:

- Translation between different codecs
- Translation between different packetization settings (transrating)
- DTMF interworking

**Transcoding the Distributed SBC**

Transcoding is inferred from a Session Description Protocol (SDP) that is used to program a call. Programming terminations in the same call containing different codecs implicitly instruct the distributed SBC to perform transcoding.
Transrating the Distributed SBC

Transrating is inferred from the SDP that is used to program a call. Programming terminations in the same call with different ptime implicitly instruct the distributed SBC to perform transrating.

**Note**

Transrating is supported only for the different rates using the same codec, not across codecs. Therefore, transrating and transcoding cannot be performed simultaneously.

---

**RTP Telephone-Event Codec-to-SIP Interworking**

When an RTP packet is marked as DTMF using the telephone-event codec, the RTP packet is removed from the stream. The DBE sends an H.248 message to the signaling border element (SBE), indicating that a DTMF event has occurred, and that the RTP packet should be converted into a SIP DTMF event.

The call must meet the following conditions:

- The telephone-event codec (for RFC 2833) is present in side A of the SDP, but not in side B.
- The dd/etd event is subscribed for side A, but not for side B.

---

**SIP-to-RTP Telephone-Event Codec Interworking**

When an endpoint generates a SIP signal, the SIP DTMF signals arrive completely out of band. An endpoint that supports SIP DTMF generates the signals to be sent to the SBE. In turn, the SBE recognizes that this is a DTMF message and sends an H.248 message to the DBE, indicating that a DTMF tone is required to be inserted into the RTP stream. The DBE then inserts the RTP DTMF packets into the audio stream using telephone-event codec.

The call must meet the following conditions:

- The telephone-event codec (for RFC 2833) is present in side B of the SDP, but not in side A.
- The dd/etd event is subscribed for side B, but not for side A.

---

**RTP Telephone-Event Codec-to-RTP In-Band Waveform**

After the RTP packet is marked as DTMF using the telephone-event codec, the RTP packet is removed from the stream, and an RTP stream containing the DTMF waveform is sent to the other endpoint.

The call must meet the following conditions:

- The telephone-event codec (for RFC 2833) is present in side A of SDP, but not in side B.
- The dd/etd event is subscribed for side A and side B.

---

**RTP In-Band Waveform-to-RTP Telephone-Event Codec**

After the DTMF is sent as part of the voice waveform, the RTP packets are removed from the stream, and the DBE inserts a new RTP packet with the payload-type telephone event into the audio stream.

The call must meet the following conditions:

- The telephone-event codec (for RFC 2833) is present in side B of the SDP, but not in side A.
- The dd/etd event is subscribed for side A and side B.
SIP-to-RTP In-Band Waveform

After an endpoint generates a SIP signal, the SIP DTMF signals arrive completely out of band. The endpoint that supports SIP DTMF generates the signals to be sent to the SBE. In turn, the SBE recognizes that this is a DTMF message, and sends an H.248 message to the DBE, indicating that a DTMF tone is required to be inserted into the RTP stream. The DBE then inserts a stream containing the DTMF waveform.

The call must meet the following conditions:

- The telephone-event codec (for RFC 2833) is not present on either side A or side B.
- The dd/etd event is subscribed for side B.

RTP In-Band Waveform-to-SIP

When the DTMF is sent as part of the voice waveform, the RTP packets are removed from the stream, and the DBE sends an H.248 message to the SBE, indicating that a DTMF event has occurred, and that the RTP packets should be converted into a SIP DTMF event.

The call must meet the following conditions:

- The telephone-event codec (for RFC 2833) is not present on either side A or side B.
- The dd/etd event is subscribed for side A.

Call Recovery

From Cisco IOS XE Release 3.3S, calls on a partially crashed SPA DSP can be recovered within the call outage time of 2.5s.

When part of a SPA DSP crashes, a crash recovery process runs, and then the RP reprograms the crashed part of the SPA DSP with all calls that were previously on it. For example, a simple transcoding scenario, a-law to u-law transcoding, can represent up to 129 calls that require reprogramming.

Depending on the part of the SPA DSP that crashes, the total recovery time may be longer because it might have to recover more components and also reprogram more calls. However, the entire media path outage time for all the recovered calls is less than 2.5s.

In all cases of the SPA DSP call recovery, the call recovery occurs on the same SPA DSP where the call existed prior to the crash. The calls are not moved to another SPA DSP.

The SPA DSP failure call recovery can be disabled or rendered ineffective if the SPA DSP crash dumps are enabled. It can push the call outage time beyond 2.5s.

The `show voice dsp group all` command indicates when a SPA DSP is undergoing call recovery.

Router# show voice dsp group all

Show DSP group all

DSP groups on slot 0 bay 0:
dsp 1:
  State: UP
  HA State : DSP_HA_STATE_PENDING1
  Max signal/voice channel: 43/43
  Max credits: 645
  num_of_sig_chnls_allocated: 43
  Transcoding channels allocated: 43
  Group: FLEX_GROUP_XCODE, complexity: LOW
Note

The show voice dsp group all command displays the output HA State: DSP_HA_STATE_PENDING only during the recovery process which can be up to a few milliseconds.

AMR-WB Transcoding Support

Adaptive Multi-Rate Wideband (AMR-WB) is a patented speech coding standard based on Adaptive Multi-Rate encoding, using a methodology that is similar to the Algebraic code-excited linear prediction (ACELP). AMR-WB, which was specified by 3GPP, provides improved speech quality due to a wider speech bandwidth of 50 to 7000Hz compared to narrowband speech coders which are in general optimized for Plain old telephone service (POTS) wireline quality of 300 to 3400 Hz.

AMR-WB is codified as G.722.2, an ITU-T standard speech codec, formally known as Wideband coding of speech at around 16 kbps using AMR-WB. G.722.2 AMR-WB is the same codec as the 3GPP AMR-WB.

AMR-WB operates like AMR with nine different bit rates. The lowest bit rate providing excellent speech quality in a clean environment is 12.65 kbps. Higher bit rates are useful in background noise conditions and for music. Also, lower bit rates of 6.60 and 8.85 kbps provide reasonable quality, especially compared to narrowband codecs.

Note

The AMR-WB feature requires DSP firmware with AMR-WB codec support.
Table 41-1 shows the relationship between the AMR rate mode and bit-rate.

<table>
<thead>
<tr>
<th>Rate Mode</th>
<th>AMR Bit-Rate (kbps)</th>
<th>AMR-WB/G.722.2 Bit-Rate (kbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4.75</td>
<td>6.60</td>
</tr>
<tr>
<td>1</td>
<td>5.15</td>
<td>8.85</td>
</tr>
<tr>
<td>2</td>
<td>5.90</td>
<td>12.65</td>
</tr>
<tr>
<td>3</td>
<td>6.70</td>
<td>14.25</td>
</tr>
<tr>
<td>4</td>
<td>7.40</td>
<td>15.85</td>
</tr>
<tr>
<td>5</td>
<td>7.95</td>
<td>18.25</td>
</tr>
<tr>
<td>6</td>
<td>10.20</td>
<td>19.85</td>
</tr>
<tr>
<td>7</td>
<td>12.20</td>
<td>23.05</td>
</tr>
<tr>
<td>8</td>
<td>SID$^{1}$</td>
<td>23.85</td>
</tr>
<tr>
<td>9</td>
<td>—</td>
<td>SID</td>
</tr>
</tbody>
</table>

1. SID: Silence Indicator

**Configuring the SPA DSP Services for SBC**

This section describes the tasks to involved in configuring the SPA DSP services for the SBC:

- Setting Up a SPA DSP for DSP Farm Services, page 41-7
- Configuring a DSP Farm Profile, page 41-8

**Setting Up a SPA DSP for DSP Farm Services**

Use the following procedure to set up the SPA DSP in the DSP farm mode for the DSP services:

**SUMMARY STEPS**

1. `configure terminal`
2. `voice-card slot number/subslot number`
3. `dsp services dspfarm`
4. `end`
DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 2 voice-card slot number/subslot number</td>
<td>Specifies the slot number of the voice card and enters the voice card interface configuration mode.</td>
</tr>
<tr>
<td>Example: Router(config)# voice-card 0/2</td>
<td></td>
</tr>
<tr>
<td>Step 3 dsp services dspfarm</td>
<td>Allows DSP farm services on the SPA DSP voice card.</td>
</tr>
<tr>
<td>Example: Router(config-voicecard)# dsp services dspfarm</td>
<td></td>
</tr>
<tr>
<td>Step 4 end</td>
<td>Exits the voice card interface configuration mode.</td>
</tr>
<tr>
<td>Example: Router(config-voicecard)# end</td>
<td></td>
</tr>
</tbody>
</table>

For more information about configuring DSP farm services on a SPA DSP, see the “Configuring the Cisco DSP SPA for ASR 1000 Series” chapter in the Cisco ASR 1000 Series Aggregation Services Routers SIP and SPA Software Configuration Guide at:


**Configuring a DSP Farm Profile**

Use the following steps to configure a DSP farm profile:

**SUMMARY STEPS**

1. configure terminal
2. dspfarm profile profile-identifier {conference | mtp | transcode}
3. description profile-description-text
4. codec codec-name
5. associate application {cube | sbc | sccp}
6. maximum session number
7. no shutdown
8. end
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters the global configuration mode.</td>
</tr>
</tbody>
</table>
| **Step 2** dspfarm profile profile-identifier {conference | mtp | transcode} | Enables the DSP farm service for the specified DSP farm profile, and enters a DSP farm profile configuration mode. The service options are:  
  - **conference**—Enables conferencing.  
  - **mtp**—Enables media termination point.  
  - **transcode**—Enables transcoding of information.  
  **Note** In Cisco IOS Release 3.2S, only the transcode service is supported. |
| **Step 3** description profile-description-text | Specifies a description for a defined profile. |
| **Step 4** no codec codec-name | Adds codecs or removes the codec from a codec list. The codec must be present in the list of codecs that the SBE is hard-coded to recognize. |
| **Step 5** associate application {cube | sbc | sccp}; profile-description-text | Associates an application to the profile. The applications that can be associated are:  
  - **cube**—Associates the Cisco Unified Border Element application to a defined profile in the DSP farm.  
  - **sbc**—Associates the SBC application to a defined profile in the DSP farm.  
  - **sccp**—Associates the client control protocol application to a defined profile in the DSP farm.  
  **Note** The sbc application keyword is available only when a DSP farm profile transcode service is used. |
Configuring the Unified SBC

This section explains the various ways in which to configure the SBC for the SPA DSP voice card:

- Associating the Unified SBC with a DSP Farm Profile, page 41-10
- Configuring the Unified SBC to Enable Transcoding, page 41-11
- Configuring the Unified SBC to Enable Transrating, page 41-17
- Configuring the Unified SBC to Enable SRTP and Transcoding, page 41-22
- Configuring the Unified SBC for Inband DTMF Interworking, page 41-28
- Configuring the Unified SBC to Support AMR-WB, page 41-33

Associating the Unified SBC with a DSP Farm Profile

Association of the SBC to the DSP farm profiles is possible only after the corresponding DSP farm profile is created. Use the `associate dspfarm profile` command in the global configuration mode.

**SUMMARY STEPS**

1. `show dspfarm {all | dsp | profile}
2. configure terminal
3. `sbc sbc-name
4. `associate dspfarm profile {profile-number | all}
5. `end

### Step 6

**maximum session number**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-dspfarm-profile)# maximum session 300</code></td>
<td>Establishes the maximum number of sessions that can be assigned to a defined profile. The maximum number of sessions is dependent upon the number of SPA DSPs in the router, and the codecs configured. For a fully populated Cisco ASR 1013 Series Router with 23 SPA DSPs and only the G711 codec, the maximum number of sessions would be 20769.</td>
</tr>
</tbody>
</table>

### Step 7

**no shutdown**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-dspfarm-profile)# no shutdown</code></td>
<td>Enables or disables a DSP farm profile.</td>
</tr>
</tbody>
</table>

### Step 8

**end**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-dspfarm-profile)# end</code></td>
<td>Exits the DSP farm profile.</td>
</tr>
</tbody>
</table>
DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> show dspfarm (all</td>
<td>dsp</td>
</tr>
<tr>
<td>Example:</td>
<td>Router# show dspfarm profile all</td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enables the global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router# configure terminal</td>
</tr>
<tr>
<td><strong>Step 3</strong> sbc sbc-name</td>
<td>Creates the SBC service on the SBC, and enters the SBC configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config)# sbc mySBC</td>
</tr>
<tr>
<td><strong>Step 4</strong> associate dspfarm profile (profile-number</td>
<td>all)</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc)# associate dspfarm profile 1</td>
</tr>
<tr>
<td><strong>Step 5</strong> end</td>
<td>Exits the configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe)# end</td>
</tr>
</tbody>
</table>

Configuring the Unified SBC to Enable Transcoding

This task configures the SBC for enabling the transcoding feature.

Note

The caller and callee commands have been used in this procedure. In some scenarios, the branch command can be used as an alternative to the caller and callee command pair. The branch command has been introduced in Release 3.5.0. See the ?paranum>Configuring Directed Nonlimiting CAC Policies? section on page 7-37 for information about this command.

SUMMARY STEPS

1. configure terminal
2. sbc sbc-name
3. sbe
4. cac-policy-set policy-set-id
5. first-cac-scope scope-name
6. first-cac-table table-name
7. cac-table table-name
8. table-type limit list of limit tables
9. entry entry-id
10. match-value key
11. callee-codec-list list-name
12. caller-codec-list list-name
13. media police strip | reject | degrade
14. action cac-complete
15. complete
16. cac-policy-set global cac-policy-num
17. codec-list list-name
18. codec codec-nam
19. exit
20. codec-list list-name
21. codec codec-nam
22. exit
23. end
24. show sbc sbc-name sbc call-stats global current5min

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure terminal</td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td>Example: Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 2 sbc sbc-name</td>
<td>Creates the SBC service on the SBC, and enters the SBC configuration mode.</td>
</tr>
<tr>
<td>Example: Router(config)# sbc mySBC</td>
<td></td>
</tr>
<tr>
<td>Step 3 sbe</td>
<td>Enters the signaling border element (SBE) function mode of the SBC.</td>
</tr>
<tr>
<td>Example: Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td>Step 4 cac-policy-set policy-set-id</td>
<td>Enters the CAC policy set configuration mode within an SBE entity, creating a new policy set, if necessary:</td>
</tr>
<tr>
<td>Example: Router(config-sbc-sbe)# cac-policy-set 1</td>
<td>• policy-set-id—Integer chosen by a user to identify the policy set. The range is from 1 to 2147483647.</td>
</tr>
</tbody>
</table>
### Command or Action

**Step 5**

```
first-cac-scope scope-name
```

**Example:**

```
Router(config-sbc-sbe-cacpolicy) #
first-cac-scope dst-adjacency
```

**Purpose:** Configures the scope at which limits should be initially defined to perform tasks at the admission control stage of the policy. Each CAC policy has a scope that can be applied to it. This CAC policy is applicable on a per call basis.  

**scope-name** has one of the following values:

- **adj-group**—Limits for events from members of the same adjacency group.
- **call**—Limits are per single call.
- **category**—Limits per category.
- **dst-account**—Limits for events sent to the same account.
- **dst-adj-group**—Limits for events sent to the same adjacency group.
- **dst-adjacency**—Limits for events sent to the same adjacency.
- **dst-number**—Limits for events that have the same adjacency number.
- **global**—Limits are global and should not be combined with any other option.
- **src-account**—Limits for events from the same account.
- **src-adj-group**—Limits for events from the same adjacency group.
- **src-adjacency**—Limits for events from the same adjacency.
- **src-number**—Limits for events that have the same source number.

**Step 6**

```
first-cac-table table-name
```

**Example:**

```
Router(config-sbc-sbe-cacpolicy) #
first-cac-table codec-dst-acc
```

**Purpose:** Configures the name of the first policy table to be processed. A CAC policy may have many tables configured. To start applying the CAC policy, the first table that is used must be defined:

- **table-name**—The admission control table that should be processed first.

**Step 7**

```
cac-table table-name
```

**Example:**

```
Router(config-sbc-sbe-cacpolicy) # cac-table
codec-dst-acc
```

**Purpose:** Enters the CAC table mode for configuration of an admission control table (creating one, if necessary) within the context of an SBE policy set.

- **table-name**—Name of the admission control table.
### Command or Action

#### Step 8  
**table-type limit list of limit tables**

**Example:**
```
Router(config-sbc-sbe-cacpolicy-cactable)#
table-type limit dst-adjacency
```

#### Step 9  
**entry entry-id**

**Example:**
```
Router(config-sbc-sbe-cacpolicy-cactable)#
entry 1
```

#### Step 10  
**match-value key**

**Example:**
```
Router(config-sbc-sbe-cacpolicy-cactable-entry)#
match-value nava
```

#### Step 11  
**callee-codec-list list-name**

**Example:**
```
Router(config-sbc-sbe-cacpolicy-cactable-entry)#
callee-codec-list PCMU
```

### Purpose

- **Configures a new CAC Limit table type in which the criteria used to match the entries must be entered.**
  - `list of limit tables` can be one of the following values:
    - **account**—Compare the name of the account.
    - **adj-group**—Compare the name of the adjacency group.
    - **adjacency**—Compare the name of the adjacency.
    - **all**—No comparison type. All events match this type.
    - **call-priority**—Compare with call priority.
    - **category**—Compare the number analysis assigned category.
    - **dst-account**—Compare the name of the destination account.
    - **dst-adj-group**—Compare the name of the destination adjacency group.
    - **dst-adjacency**—Compare the name of the destination adjacency.
    - **dst-prefix**—Compare the beginning of the dialed digit string.
    - **event-type**—Compare with CAC policy event types.
    - **src-account**—Compare the name of the source account.
    - **src-adj-group**—Compare the name of the source adjacency group.
    - **src-adjacency**—Compare the name of the source adjacency.
    - **src-prefix**—Compare the beginning of the calling number string.

- **Enters the CAC table entry mode to modify an entry in an admission control table.**
  - `entry-id`—Specifies the table entry.

- **Configures the match value of an entry in a CAC Limit table type.**

- **Lists the codecs that the callee leg of a call is allowed to use.**
<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td><code>caller-codec-list list-name</code></td>
<td>Lists the codecs that the caller leg of a call is allowed to use.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
</tbody>
</table>
|      | `Router(config-sbc-sbe-cacpolicy-cactable-entry)
# caller-codec-list PCMA`                           |                                                                                                                                          |
| 13   | `media police strip | reject | degrade`                                                                               | Configures the manner in which the SBC will handle the media streams that exceed the bandwidth limit for media calls.                  |
|      | **Example:**                               |                                                                                                                                          |
|      | `Router(config-sbc-sbe-cacpolicy-cactable-entry)
# media police strip`                           |                                                                                                                                          |
| 14   | `action cac-complete`                       | When an event matches, the CAC policy is considered complete.                                                                             |
|      | **Example:**                               |                                                                                                                                          |
|      | `Router(config-sbc-sbe-cacpolicy-cactable-entry)
# action cac-complete`                           |                                                                                                                                          |
| 15   | `complete`                                 | Completes the CAC policy set when you have committed the full set.                                                                            |
|      | **Example:**                               |                                                                                                                                          |
|      | `Router(config-sbc-sbe-cacpolicy)# complete`                           |                                                                                                                                          |
| 16   | `cac-policy-set global policy-num`         | Activates the global CAC policy set. The CAC policy set must be in a complete state before it can be assigned as the default policy.        |
|      | **Example:**                               |                                                                                                                                          |
|      | `Router(config-sbc-sbe)# cac-policy-set global 1`                           | • *policy-num*—The call policy set number, ranging from 1 to 2147483647. The policy set must be in a complete state before it can be assigned as the default policy. |
| 17   | `codec-list list-name`                     | Creates a codec list, and enters the Codec list configuration mode.                                                                        |
|      | **Example:**                               |                                                                                                                                          |
|      | `Router(config-sbc-sbe)# codec-list PCMU`                           |                                                                                                                                          |
| 18   | `codec codec-name`                         | Adds a codec to a codec list.                                                                                                             |
|      | **Example:**                               |                                                                                                                                          |
|      | `Router(config-sbc-sbe-codec-list)# codec PCMU`                           |                                                                                                                                          |
| 19   | `exit`                                     | Exits the codec list configuration mode.                                                                                                  |
|      | **Example:**                               |                                                                                                                                          |
|      | `Router(config-sbc-sbe-codec-list)# exit`                           |                                                                                                                                          |
| 20   | `codec-list list-name`                     | Creates a codec list, and enters the Codec list configuration mode.                                                                        |
|      | **Example:**                               |                                                                                                                                          |
|      | `Router(config-sbc-sbe)# codec-list PCMA`                           |                                                                                                                                          |
| 21   | `codec codec-name`                         | Adds a codec to a codec list.                                                                                                             |
|      | **Example:**                               |                                                                                                                                          |
|      | `Router(config-sbc-sbe-codec-list)# codec PCMA`                           |                                                                                                                                          |
## Configuring the Unified SBC

### Command or Action

<table>
<thead>
<tr>
<th>Step 22</th>
<th>exit</th>
</tr>
</thead>
</table>

**Example:**

Router(config-sbc-sbe-codec-list)# exit

### Step 23 | end |

**Example:**

Router(config-sbc-sbe)# end

### Step 24 | show sbc sbc-name sbe call-stats global current5min |

**Example:**

Router# show sbc mySBC sbe call-stats global current5min

### The following example shows an output of the `show sbc sbc-name sbe call-stats global current5min` command that lists the count of the active transcoded and transrated calls.

```
Router# show sbc mySBC sbe call-stats global current5min

SBC Service "mySBC"
Statistics for the current 5 mins for global counters
Call count totals:
  Total call attempts = 0
  Total active calls = 1
  Total active IPv6 calls = 0
  Total activating calls = 0
  Total de-activating calls = 0
  Total active emergency calls = 0
  Total active e2 emergency calls = 0
  Total IMS rx active calls = 0
  Total IMS rx call renegotiation attempts = 0
  Total SRTTP-RTP interworked calls = 0
  Total active calls not using SRTTP = 1
  Total active transcoded calls = 1
  Total active transrated calls = 0
General call failure counters:
  Total call setup failures = 0
  Total active call failures = 0
  Total failed call attempts = 0
  Total failed calls due to update failure = 0
  Total failed calls due to resource failure = 0
  Total failed calls due to congestion = 0
  Total failed calls due to media failure = 0
  Total failed calls due to signaling failure = 0
  Total failed calls due to IMS rx setup failure = 0
  Total failed calls due to IMS rx renegotiation failure = 0
  Total failed calls due to RTP disallowed on call leg = 0
  Total failed calls due to SRTTP disallowed on call leg = 0
```
Configuring the Unified SBC to Enable Transrating

**Note**
Transrating is supported only for different rates using the same codec, not across codecs. Therefore, transrating and transcoding cannot be performed simultaneously.

This section describes how to enable transrating using either of the following methods:
- Transrating Using the Same Codec Policy, page 41-17
- Transrating Using a New Codec Policy, page 41-21

**Transrating Using the Same Codec Policy**

This task configures the SBC for enabling the transrating using the same codec policy.

**Note**
The caller and callee commands have been used in this procedure. In some scenarios, the branch command can be used as an alternative to the caller and callee command pair. The branch command has been introduced in Release 3.5.0. See the ?$paranum>Configuring Directed Nonlimiting CAC Policies? section on page 7-37 for information about this command.

**SUMMARY STEPS**

1. configure terminal
2. sbc sbc-name
3. sbe
4. cac-policy-set policy-set-id
5. first-cac-table table-name
6. cac-table table-name
7. table-type {policy-set | limit {list of limit tables}}
8. entry entry-id
9. cac-scope {list of scope options}
10. callee ptime 0-100
11. caller ptime 0-100
12. media police strip | reject | degrade
13. action cac complete
14. complete
15. cac-policy-set global cac-policy-num
16. end
## Detailed Steps

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>configure terminal</td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>sbc sbc-name</td>
<td>Creates the SBC service on the SBC, and enters the SBC configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config)# sbc mySBC</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>sbe</td>
<td>Enters the SBE function mode of the SBC.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>cac-policy-set policy-set-id</td>
<td>Enters the CAC policy set configuration mode within an SBE entity, creating a new policy set, if necessary.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe)# cac-policy-set 1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>first-cac-table table-name</td>
<td>Configures the name of the first policy table to be processed. A CAC policy may have many tables configured. To start applying the CAC policy, the first table that is used must be defined:</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-cacpolicy)# first-cac-table Transrate</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>cac-table table-name</td>
<td>Enters the CAC table mode for configuration of an admission control table (creating one, if necessary) within the context of an SBE policy set:</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-cacpolicy)# cac-table Transrate</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

**Step 7**

```
table-type {policy-set | limit (list of limit tables)}
```

**Example:**

Router(config-sbc-sbe-cacpolicy-cactable)#
table-type policy-set

**Purpose**

Configures the table type of a CAC table within the context of an SBC policy set.

*list of limit tables* can be one of the following values:

- **account**—Compare the name of the account.
- **adj-group**—Compare the name of the adjacency group.
- **adjacency**—Compare the name of the adjacency.
- **all**—No comparison type. All events match this type.
- **call-priority**—Compare with call priority.
- **category**—Compare the number analysis assigned category.
- **dst-account**—Compare the name of the destination account.
- **dst-adj-group**—Compare the name of the destination adjacency group.
- **dst-adjacency**—Compare the name of the destination adjacency.
- **dst-prefix**—Compare the beginning of the dialed digit string.
- **event-type**—Compare with CAC policy event types.
- **src-account**—Compare the name of the source account.
- **src-adj-group**—Compare the name of the source adjacency group.
- **src-adjacency**—Compare the name of the source adjacency.
- **src-prefix**—Compare the beginning of the calling number string.

Features can be enabled or disabled per adjacency group through CAC configuration the same way this is done per individual adjacencies. The adj-group table type matches either the source adjacency group or the destination adjacency group.

When the **policy-set** keyword is specified, use the **cac-scope** command to configure the scope within each of the entries in which limits are applied in a CAC Policy Set table.

**Step 8**

```
entry entry-id
```

**Example:**

Router(config-sbc-sbe-cacpolicy-cactable)#
entry 1

**Purpose**

Enters the CAC table entry mode to create or modify an entry in an admission control table.
### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td><strong>cac-scope</strong> <em>(list of scope options)</em></td>
<td>Enables the selection of a scope at which CAC limits are applied within each entry in a Policy Set table. &lt;br&gt;<strong>list of scope options</strong>—Specifies one of the following strings used to match events:  &lt;br&gt;- <strong>account</strong>—Events that are from the same account.  &lt;br&gt;- <strong>adjacency</strong>—Events that are from the same adjacency.  &lt;br&gt;- <strong>adj-group</strong>—Events that are from members of the same adjacency group.  &lt;br&gt;- <strong>call</strong>—Scope limits are per single call.  &lt;br&gt;- <strong>category</strong>—Events that have the same category.  &lt;br&gt;- <strong>dst-account</strong>—Events that are sent to the same account.  &lt;br&gt;- <strong>dst-adj-group</strong>—Events that are sent to the same adjacency group.  &lt;br&gt;- <strong>dst-adjacency</strong>—Events that are sent to the same adjacency.  &lt;br&gt;- <strong>dst-number</strong>—Events that have the same destination.  &lt;br&gt;- <strong>global</strong>—Scope limits are global.  &lt;br&gt;- <strong>src-account</strong>—Events that are from the same account.  &lt;br&gt;- <strong>src-adj-group</strong>—Events that are from the same adjacency group.  &lt;br&gt;- <strong>src-adjacency</strong>—Events that are from the same adjacency.  &lt;br&gt;- <strong>src-number</strong>—Events that have the same source number.</td>
</tr>
<tr>
<td>10</td>
<td><strong>callee ptime</strong> <em>&lt;0-100&gt;</em></td>
<td>Configures the packetization time on the callee side that is forced for calls using this CAC entry. &lt;br&gt;By default, 0 ms is configured, which means no transrating occurs.</td>
</tr>
<tr>
<td>11</td>
<td><strong>caller ptime</strong> <em>&lt;0-100&gt;</em></td>
<td>Configures the packetization time on the caller side that is forced for calls using this CAC entry. &lt;br&gt;By default, 0 ms is configured, which means no transrating occurs.</td>
</tr>
<tr>
<td>12</td>
<td>**media police strip</td>
<td>reject</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> Router(config-sbc-sbe-cacpolicy-cactable-entry) # media police strip  # media police strip</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td><strong>action cac-complete</strong></td>
<td>When an event matches, this CAC policy is complete.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> Router(config-sbc-sbe-cacpolicy-cactable-entry) # action cac complete</td>
<td></td>
</tr>
</tbody>
</table>
Configuring the Unified SBC

Transrating Using a New Codec Policy

This task configures the SBC for enabling the transrating feature. This is an alternative mechanism to that described in the "Transrating Using the Same Codec Policy? section on page 41-17 section for configuring transrating.

SUMMARY STEPS

1. configure terminal
2. sbc sbc-name
3. sbe
4. codec list list-name
5. policy {minimum | transrating}
6. codec codec-name packetization-period packet-period [priority priority-value]
7. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td>Example: Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc sbc-name</td>
<td>Creates the SBC service on the SBC, and enters into the SBC configuration mode.</td>
</tr>
<tr>
<td>Example: Router(config)# sbc mySBC</td>
<td></td>
</tr>
</tbody>
</table>
Configuring the Unified SBC to Enable SRTP and Transcoding

Although Secure Real-time Transport Protocol (SRTP) is independent of transcoding, both can be configured to be used simultaneously.

This task configures the unified SBC to enable the SRTP and transcoding features.

Note

The `caller` and `callee` commands have been used in this procedure. In some scenarios, the `branch` command can be used as an alternative to the `caller` and `callee` command pair. The `branch` command has been introduced in Release 3.5.0. See the ?$paranum>Configuring Directed Nonlimiting CAC Policies? section on page 7-37 for information about this command.

SUMMARY STEPS

1. configure terminal
2. `sbc sbc-name`
3. `sbe`
4. `cac-policy-set policy-set-id`
5. `first-cac-table table-name`
6. `cac-table table-name`
7. `table-type {policy-set | limit {list of limit tables}}`
8. `entry entry-id`
9. `cac-scope {list of scope options}`
10. `srtp support allow`
11. `srtp caller forbid | mandate | allow | prefer`
12. `srtp callee forbid | mandate | allow | prefer`
13. `srtp interworking forbid | allow`
14. `srtp media interworking forbid | allow`
15. `action next-table goto-table-name`
16. `exit`
17. `exit`
18. `cac-table table-name`
19. `table-type limit list of limit tables`
20. `entry entry-id`
21. `match-value key`
22. `callee-codec-list list-name`
23. `action cac-complete`
24. `complete`
25. `cac-policy-set global cac-policy-num`
26. `end`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc sbc-name</td>
<td>Creates the SBC service on the SBC, and enters into the SBC configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config)# sbc mySBC</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Enters the SBE function mode of the SBC.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> cac-policy-set policy-set-id</td>
<td>Enters the CAC policy set configuration mode within an SBE entity, creating a new policy set, if necessary.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe)# cac-policy-set 3</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>---------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td></td>
</tr>
<tr>
<td><code>first-cac-table table-name</code></td>
<td>Configures the name of the first policy table to be processed. A CAC policy may have many tables configured. To start applying the CAC policy, the first table that is used must be defined:</td>
</tr>
<tr>
<td>Example:</td>
<td><code>Router(config-sbc-sbe-cacpolicy)# first-cac-table C3</code></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td></td>
</tr>
<tr>
<td><code>cac-table table-name</code></td>
<td>Enters the CAC table mode for configuration of an admission control table (creating one, if necessary) within the context of an SBE policy set:</td>
</tr>
<tr>
<td>Example:</td>
<td><code>Router(config-sbc-sbe-cacpolicy)# cac-table C3</code></td>
</tr>
</tbody>
</table>
### Command or Action

**Step 7**

`table-type {policy-set | limit (list of limit tables)}`

*Example:*

Router(config-sbc-sbe-cacpolicy-cactable)#
table-type policy-set

**Purpose**

Configures the table type of a CAC table within the context of an SBC policy set.

*list of limit tables* can be one of the following values:
- `account`—Compare the name of the account.
- `adj-group`—Compare the name of the adjacency group.
- `adjacency`—Compare the name of the adjacency.
- `all`—No comparison type. All events match this type.
- `call-priority`—Compare with call priority.
- `category`—Compare the number analysis assigned category.
- `dst-account`—Compare the name of the destination account.
- `dst-adj-group`—Compare the name of the destination adjacency group.
- `dst-adjacency`—Compare the name of the destination adjacency.
- `dst-prefix`—Compare the beginning of the dialed digit string.
- `event-type`—Compare with CAC policy event types.
- `src-account`—Compare the name of the source account.
- `src-adj-group`—Compare the name of the source adjacency group.
- `src-adjacency`—Compare the name of the source adjacency.
- `src-prefix`—Compare the beginning of the calling number string.

Features can be enabled or disabled per adjacency group through CAC configuration the same way this is done per individual adjacency. The adj-group table type matches on either the source adjacency group or the destination adjacency group.

When the policy-set keyword is specified, use the `cac-scope` command to configure the scope within each entry in which limits are applied in a CAC Policy Set table.

**Step 8**

`entry entry-id`

*Example:*

Router(config-sbc-sbe-cacpolicy-cactable)#
entry 1

**Purpose**

Enters the mode to create or modify an entry in an admission control table.
### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td><code>cac-scope (list of scope options)</code></td>
<td>Choose a scope at which CAC limits are applied within each entry in a Policy Set table.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td><strong>list of scope options</strong>—Specifies one of the following strings used to match events:</td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-cacpolicy-cactable-entry)</td>
<td>• account—Events that are from the same account.</td>
</tr>
<tr>
<td></td>
<td># cac-scope global</td>
<td>• adjacency—Events that are from the same adjacency.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• adj-group—Events that are from members of the same adjacency group.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• call—Scope limits are per single call.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• category—Events that have the same category.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• dst-account—Events that are sent to the same account.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• dst-adj-group—Events that are sent to the same adjacency group.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• dst-adjacency—Events that are sent to the same adjacency.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• dst-number—Events that have the same destination.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• global—Scope limits are global.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• src-account—Events that are from the same account.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• src-adj-group—Events that are from the same adjacency group.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• src-adjacency—Events that are from the same adjacency.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• src-number—Events that have the same source number.</td>
</tr>
<tr>
<td>10</td>
<td><code>srtp support allow</code></td>
<td>Configures SRTP support.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-cacpolicy-cactable-entry)</td>
</tr>
<tr>
<td></td>
<td># srtp support allow</td>
<td><strong>Step 11</strong> **srtp caller forbid</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Configures SRTP for the caller side of the call with one of the following SRTP settings:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• forbid—SRTP is not supported on the caller side of the call.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• mandate—SRTP is mandatory on the caller side of the call.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• allow—SRTP is optional on the caller side of the call.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• prefer—SRTP is preferred on this adjacency. Both RTP and SRTP are accepted inbound, but only SRTP is offered outbound.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-cacpolicy-cactable-entry)</td>
</tr>
<tr>
<td></td>
<td># srtp caller mandate</td>
<td><strong>Step 11</strong> **srtp caller forbid</td>
</tr>
</tbody>
</table>
### Command or Action

#### Step 12
`srtpt callee forbid | mandate | allow | prefer`

**Example:**
```
Router(config-sbc-sbe-cacpolicy-cactable-entry)
# srtpt callee mandate
```

**Purpose:** Configures SRTP for the callee side of the call:
- *forbid*—SRTP is not supported on the callee side of the call.
- *mandate*—SRTP is mandatory on the callee side of the call.
- *allow*—SRTP is optional on the callee side of the call.
- *prefer*—SRTP is preferred on this adjacency. Both RTP and SRTP are accepted inbound, but only SRTP is offered outbound.

#### Step 13
`srtpt interworking forbid | allow`

**Example:**
```
Router(config-sbc-sbe-cacpolicy-cactable-entry)
# srtpt interworking allow
```

**Purpose:** Configures SRTP-to-RTP interworking.
- *forbid*—Prohibits SRTP-to-RTP interworking on a call.
- *allow*—Allows SRTP-to-RTP interworking on a call.

#### Step 14
`srtpt media interworking forbid | allow`

**Example:**
```
Router(config-sbc-sbe-cacpolicy-cactable-entry)
# srtpt media interworking allow
```

**Purpose:** Configures SRTP-to-RTP media interworking.
- *forbid*—Prohibits SRTP-to-RTP media interworking on a call.
- *allow*—Allows SRTP-to-RTP media interworking on a call.

#### Step 15
`action next-table goto-table-name`

**Example:**
```
Router(config-sbc-sbe-cacpolicy-cactable-entry)
# action next-table xcode
```

**Purpose:** Configures the action to be taken when the routing entry is chosen.
- *goto-table-name*—Specifies the next routing table to be processed when an event matches the entry.

#### Step 16
`exit`

**Example:**
```
Router(config-sbc-sbe-cacpolicy-cactable-entry)
# exit
```

**Purpose:** Exits the CAC table entry configuration mode.

#### Step 17
`exit`

**Example:**
```
Router(config-sbc-sbe-cacpolicy-cactable)# exit
```

**Purpose:** Exits the CAC table configuration mode.

#### Step 18
`cac-table table-name`

**Example:**
```
Router(config-sbc-sbe-cacpolicy-cactable)# cac-table xcode
```

**Purpose:** Enters the CAC table mode for configuration of an admission control table (creating one, if necessary) within the context of an SBE policy set.
- *table-name*—Name of the admission control table.

#### Step 19
`table-type limit list of limit tables`

**Example:**
```
Router(config-sbc-sbe-cacpolicy-cactable)# table-type limit dst-adjacency
```

**Purpose:** Configures the limit of the table types to be matched by the *match-value* command. For the example provided here, use the following table type:
- *dst-adjacency*—Compares the name of the destination adjacency.
**Command or Action** | **Purpose**
--- | ---
**Step 20** | **entry entry-id**
- **Example:**
  ```
  Router(config-sbc-sbe-cacpolicy-cactable)# entry 1
  ```
- Enters the CAC table entry mode to modify an entry in an admission control table.
  - **entry-id**—Specifies the table entry.

**Step 21** | **match-value key**
- **Example:**
  ```
  Router(config-sbc-sbe-cacpolicy-cactable-entry)# match-value nav4B
  ```
- Configures the match-value of an entry in a Call Admission Control (CAC) Limit table:
  - **key**—Specifies the keyword used to match events. The format of the key is determined by the table-type limit.

**Step 22** | **callee-codec-list list-name**
- **Example:**
  ```
  Router(config-sbc-sbe-cacpolicy-cactable-entry)# callee-codec-list PCMU
  ```
- Lists the codecs that the callee leg of a call is allowed to use:
  - **list-name**—Specifies the name of the codec list. The maximum size is 30 characters.

**Step 23** | **action cac-complete**
- **Example:**
  ```
  Router(config-sbc-sbe-cacpolicy-cactable-entry)# action cac-complete
  ```
- When the event matches, this CAC policy is complete.

**Step 24** | **complete**
- **Example:**
  ```
  Router(config-sbc-sbe-cacpolicy)# complete
  ```
- Completes the CAC policy set when you have committed the full set.

**Step 25** | **cac-policy-set global policy-num**
- **Example:**
  ```
  Router(config-sbc-sbe-cacpolicy)# cac-policy-set global 3
  ```
- Activates the global CAC policy set. The CAC policy set must be in a complete state before it can be assigned as the default policy.
  - **policy-num**—The call policy set number, ranging from 1 to 2147483647. The policy set must be in a complete state before it can be assigned as the default policy.

**Step 26** | **end**
- **Example:**
  ```
  Router(config-sbc-sbe-cacpolicy-cactable-entry)# end
  ```
- Exits the CAC configuration mode and returns to privileged EXEC mode.

### Configuring the Unified SBC for Inband DTMF Interworking

A SPA DSP can be used to detect the DTMF tones, called inband, that are played in the real-time transport protocol (RTP) stream. Inband DTMF interworking uses SPA DSP resources, and can be used for plain calls and transcoded calls.
Note
The `caller` and `callee` commands have been used in this procedure. In some scenarios, the `branch` command can be used as an alternative to the `caller` and `callee` command pair. The `branch` command has been introduced in Release 3.5.0. See the “Configuring Directed Nonlimiting CAC Policies” section on page 7-37 for information about this command.

SUMMARY STEPS

1. configure terminal
2. sbc sbc-name
3. sbe
4. cac-policy-set policy-set-id
5. first-cac-table table-name
6. cac-table table-name
7. table-type { policy-set | limit { list of limit tables } }
8. entry entry-id
9. cac-scope { list of scope options }
10. callee inband-dtmf-mode always
11. caller inband-dtmf-mode never
12. action next-table goto-table-name
13. complete
14. cac-policy-set global cac-policy-num
15. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure terminal</td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td>Example: Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 2 sbc sbc-name</td>
<td>Creates the SBC service on the SBC, and enters into the SBC configuration mode.</td>
</tr>
<tr>
<td>Example: Router(config)# sbc mySBC</td>
<td></td>
</tr>
<tr>
<td>Step 3 sbe</td>
<td>Enters the SBE function mode of the SBC.</td>
</tr>
<tr>
<td>Example: Router(config-sbc)# sbe</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Step 4</th>
<th>cac-policy-set policy-set-id</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe)# cac-policy-set 4</td>
</tr>
</tbody>
</table>

- **Purpose**: Enters the CAC policy set configuration mode within an SBE entity, creating a new policy set, if necessary.

<table>
<thead>
<tr>
<th>Step 5</th>
<th>first-cac-table table-name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-cacpolicy)# first-cac-table C4</td>
</tr>
</tbody>
</table>

- **Purpose**: Configures the name of the first policy table to be processed. A CAC policy may have many tables configured. To start applying the CAC policy, the first table that is used must be defined:
  - **table-name**—The admission control table that should be processed first.

<table>
<thead>
<tr>
<th>Step 6</th>
<th>cac-table table-name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-cacpolicy)# cac-table C4</td>
</tr>
</tbody>
</table>

- **Purpose**: Enters the CAC table mode for configuration of an admission control table (creating one, if necessary) within the context of an SBE policy set:
  - **table-name**—Name of the admission control table.
### Command or Action

**Step 7**  
```plaintext
table-type {policy-set | limit (list of limit tables)}
```

**Example:**  
```plaintext
Router(config-sbc-sbe-cacpolicy-cactable)#
table-type policy-set
```

**Purpose**  
Configures the table type of a CAC table within the context of an SBC policy set.  

**list of limit tables** can be one of the following values:  
- **account**—Compare the name of the account.  
- **adj-group**—Compare the name of the adjacency group.  
- **adjacency**—Compare the name of the adjacency.  
- **all**—No comparison type. All events match this type.  
- **call-priority**—Compare with call priority.  
- **category**—Compare the number analysis assigned category.  
- **dst-account**—Compare the name of the destination account.  
- **dst-adj-group**—Compare the name of the destination adjacency group.  
- **dst-adjacency**—Compare the name of the destination adjacency.  
- **dst-prefix**—Compare the beginning of the dialed digit string.  
- **event-type**—Compare with CAC policy event types.  
- **src-account**—Compare the name of the source account.  
- **src-adj-group**—Compare the name of the source adjacency group.  
- **src-adjacency**—Compare the name of the source adjacency.  
- **src-prefix**—Compare the beginning of the calling number string.

Features can be enabled or disabled per adjacency group through CAC configuration the same way this is done per individual adjacency. The adj-group table type matches either the source adjacency group or destination adjacency group.

When the policy-set keyword is specified, use the `cac-scope` command to configure the scope within each entry at which limits are applied in a CAC Policy Set table.

### Command or Action

**Step 8**  
```plaintext
entry entry-id
```

**Example:**  
```plaintext
Router(config-sbc-sbe-cacpolicy-cactable)#
entry 1
```

**Purpose**  
Enters the CAC table entry mode to create or modify an entry in an admission control table.
### Configuring the Unified SBC

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| 9    | `cac-scope (list of scope options)` | Choose the scope at which CAC limits are applied within each entry in a Policy Set table.  

*list of scope options*—Specifies one of the following strings used to match events:  
- *account*—Events that are from the same account.  
- *adjacency*—Events that are from the same adjacency.  
- *adj-group*—Events that are from members of the same adjacency group.  
- *call*—Scope limits are per single call.  
- *category*—Events that have the same category.  
- *dst-account*—Events that are sent to the same account.  
- *dst-adj-group*—Events that are sent to the same adjacency group.  
- *dst-adjacency*—Events that are sent to the same adjacency.  
- *dst-number*—Events that have the same destination.  
- *global*—Scope limits are global  
- *src-account*—Events that are from the same account.  
- *src-adj-group*—Events that are from the same adjacency group.  
- *src-adjacency*—Events that are from the same adjacency.  
- *src-number*—Events that have the same source number. |
| 10   | `callee inband-dtmf-mode (always | inherit | maybe | never)` | Configures the DTMF inband mode for the callee side.  

- *always*—The inband DTMF tones are always in use by the endpoint.  
- *inherit*—The inband DTMF mode for the endpoint is not affected by this CAC entry.  
- *maybe*—The inband DTMF tones are used by the endpoint unless signaling indicates that an alternative format for DTMF is in use.  
- *never*—The endpoint never uses inband DTMF. |
| 11   | `caller inband-dtmf-mode (always | inherit | maybe | never)` | Configures the DTMF inband mode for the caller side.  

- *always*—The inband DTMF tones are always in use by the endpoint.  
- *inherit*—The inband DTMF mode for the endpoint is not affected by this CAC entry.  
- *maybe*—The inband DTMF tones are used by the endpoint unless signaling indicates that an alternative format for DTMF is in use.  
- *never*—The endpoint never uses inband DTMF. |

**Example:**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| 9    | `Router(config-sbc-sbe-cacpolicy-cactable-entry) # cac-scope global` | Choose the scope at which CAC limits are applied within each entry in a Policy Set table.  

*list of scope options*—Specifies one of the following strings used to match events:  
- *account*—Events that are from the same account.  
- *adjacency*—Events that are from the same adjacency.  
- *adj-group*—Events that are from members of the same adjacency group.  
- *call*—Scope limits are per single call.  
- *category*—Events that have the same category.  
- *dst-account*—Events that are sent to the same account.  
- *dst-adj-group*—Events that are sent to the same adjacency group.  
- *dst-adjacency*—Events that are sent to the same adjacency.  
- *dst-number*—Events that have the same destination.  
- *global*—Scope limits are global  
- *src-account*—Events that are from the same account.  
- *src-adj-group*—Events that are from the same adjacency group.  
- *src-adjacency*—Events that are from the same adjacency.  
- *src-number*—Events that have the same source number. |
| 10   | `Router(config-sbc-sbe-cacpolicy-cactable-entry) # callee inband-dtmf-mode always` | Configures the DTMF inband mode for the callee side.  

- *always*—The inband DTMF tones are always in use by the endpoint.  
- *inherit*—The inband DTMF mode for the endpoint is not affected by this CAC entry.  
- *maybe*—The inband DTMF tones are used by the endpoint unless signaling indicates that an alternative format for DTMF is in use.  
- *never*—The endpoint never uses inband DTMF. |
| 11   | `Router(config-sbc-sbe-cacpolicy-cactable-entry) # caller inband-dtmf-mode never` | Configures the DTMF inband mode for the caller side.  

- *always*—The inband DTMF tones are always in use by the endpoint.  
- *inherit*—The inband DTMF mode for the endpoint is not affected by this CAC entry.  
- *maybe*—The inband DTMF tones are used by the endpoint unless signaling indicates that an alternative format for DTMF is in use.  
- *never*—The endpoint never uses inband DTMF. |
### Configuring the Unified SBC to Support AMR-WB

This section explains how to configure the Unified SBC to support AMR-WB.

#### SUMMARY STEPS

1. configure terminal
2. dspfarm profile profile-identifier transcode
3. codec amr-wb
4. sbc sbc-name
5. associate dspfarm profile profile-identifier
6. activate

---

<table>
<thead>
<tr>
<th>Step 12</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
|         | action next-table goto-table-name | Configures the action to take when this routing entry is chosen.  
|         | • goto-table-name—Specifies the next routing table to be processed when an event matches the entry. |
| Example: | Router(config-sbc-sbe-cacpolicy-cactable-entry) # action next-table xcode |

<table>
<thead>
<tr>
<th>Step 13</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>complete</td>
<td>Completes the CAC policy set when you have committed the full set.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-cacpolicy)# complete</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 14</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
|         | cac-policy-set global policy-num | Activates the global CAC policy set. The CAC policy set must be in a complete state before it can be assigned as the default policy.  
|         | • policy-num—The call policy set number, ranging from 1 to 2147483647. The policy set must be in a complete state before it can be assigned as the default policy. |
| Example: | Router(config-sbc-sbe)# cac-policy-set global 4 |

<table>
<thead>
<tr>
<th>Step 15</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>end</td>
<td>Exits the CAC configuration mode and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-cacpolicy-cactable-entry) # end</td>
<td></td>
</tr>
</tbody>
</table>
DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure terminal</td>
<td>Enters the global configuration mode.</td>
</tr>
<tr>
<td>Example: Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 2 dspfarm profile profile-identifier transcode</td>
<td>Enters the DSP farm profile configuration mode, and defines a profile for DSP farm services.</td>
</tr>
<tr>
<td>Example: Router(config)# dspfarm profile 20 transcode</td>
<td></td>
</tr>
<tr>
<td>Step 3 codec amr-wb</td>
<td>Specifies the AMR-WB codec in the DSP farm profile.</td>
</tr>
<tr>
<td>Example: Router(config-dspfarm-profile)#codec amr-wb</td>
<td></td>
</tr>
<tr>
<td>Step 4 sbc sbc-name</td>
<td>Enters the mode of an SBC service.</td>
</tr>
<tr>
<td>Example: Router(config)# sbc mySBC dbe</td>
<td></td>
</tr>
<tr>
<td>Step 5 associate dspfarm profile profile-identifier</td>
<td>Associates a DSP farm profile to a Cisco Call Manager group.</td>
</tr>
<tr>
<td>Example: Router(config-sbc-dbe)# associate profile 20</td>
<td></td>
</tr>
<tr>
<td>Step 6 activate</td>
<td>Initiates the DBE service of the SBC.</td>
</tr>
<tr>
<td>Example: Router(config-sbc-dbe)# activate</td>
<td></td>
</tr>
</tbody>
</table>

Configuration Examples of the SPA DSP Services for the SBC

This section contains the following examples:

- Example: Enabling DSP Farm Service on the SPA DSP, page 41-34
- Example: Configuring a DSP Farm Profile, page 41-35
- Example: Viewing a DSP Farm Profile Configuration and Status, page 41-35

Example: Enabling DSP Farm Service on the SPA DSP

The following example shows how to enable DSP farm services on the SPA DSP:

```
enable
configure terminal
voice-card 0/2
dsp services dspfarm
end
```
Example: Configuring a DSP Farm Profile

The following example shows how to configure a DSP farm profile:

```
enable
configure terminal
dspfarm profile 1 transcode
description enables transcoding
codec g711ulaw
codec g711alaw
codec g729ar8
codec g729abr8
codec g729r8
codec g723r63
codec ilbc
codec gsmamr-nb
codec g726r32
codec g729br8
associate application sbc
maximum session 300
end
```

Example: Viewing a DSP Farm Profile Configuration and Status

After a DSP farm profile is created, use the `show` command to display a DSP farm profile configuration and status. The following examples show the output of the `show` commands:

```
Router# show running-config
!
voice-card 2/0
no dspfarm
dsp services dspfarm
!
dspfarm profile 20 transcode
codec g711ulaw
codec g711alaw
codec g729r8
codec g729ar8
codec g729br8
codec g729abr8
associate application sbc
rsvp
maximum sessions 5
!
Router# show dspfarm profile 20
Dspfarm Profile Configuration
Profile ID = 20, Service = TRANSCODING, Resource ID = 1
Profile Description :
Profile Admin State : UP
Profile Operation State : ACTIVE
Application : SBC Status : ASSOCIATED
Resource Provider : FLEX_DSPRM Status : UP
Number of Resource Configured : 5
Number of Resource Available : 5
Codec Configuration
Codec : g729abr8, Maximum Packetization Period : 60
Codec : g711ulaw, Maximum Packetization Period : 30
Codec : g711alaw, Maximum Packetization Period : 30
Codec : g729r8, Maximum Packetization Period : 60
```
Codec: g729ar8, Maximum Packetization Period: 60
Codec: g729br8, Maximum Packetization Period: 60
RSVP: ENABLED

Router# show dspfarm all

DSPFARM Configuration Information:
Admin State: UP, Oper Status: ACTIVE - Cause code: NONE
Transcoding Sessions: 0 (Avail: 0), Conferencing Sessions: 2 (Avail: 2)
Trans sessions for mixed-mode conf: 0 (Avail: 0), RTP Timeout: 600
Connection check interval 600 Codec G729 VAD: ENABLED
Total number of active session(s) 0, and connection(s) 0
SLOT DSP CHNL STATUS USE TYPE SESS-ID CONN-ID PKTS-RXED PKTS-TXED
0 0 1 UP FREE conf- - - -
0 0 2 UP FREE conf - - - -
0 0 3 UP FREE conf - - - -
0 0 4 UP FREE conf - - - -
0 0 5 UP FREE conf - - - -
0 0 6 UP FREE conf - - - -

Configuration Examples of Unified SBC

This section contains the following examples:

- Example: Associating the Unified SBC with a DSP Farm Profile, page 41-36
- Example: Configuring the Unified SBC to Enable Transcoding, page 41-36
- Example: Configuring the Unified SBC to Enable Transrating, page 41-37
- Example: Configuring the Unified SBC to Enable SRTP and Transcoding, page 41-38
- Example: Configuring the Unified SBC for In-Band DTMF Interworking, page 41-38
- Example: Configuring the Unified SBC to Support AMR-WB, page 41-39

Example: Associating the Unified SBC with a DSP Farm Profile

The following example shows how to associate the Unified SBC with a DSP farm profile:

```
enable
configure terminal
sbc mySBC
associate dspfarm profile 1
end
```

Example: Configuring the Unified SBC to Enable Transcoding

The following example shows how to configure the unified SBC to enable transcoding.

```
Note
The caller and callee commands have been used in this procedure. In some scenarios, the branch command can be used as an alternative to the caller and callee command pair. The branch command has been introduced in Release 3.5.0. See the Paranum>Configuring Directed Nonlimiting CAC Policies? section on page 7-37 for information about this command.
```
enable
configure terminal
sbc mySBC
sbe
  cac-policy-set 1
  first-cac-scope dst-adjacency
  first-cac-table codec-dst-acc
  cac-table codec-dst-acc
  table-type limit dst-adjacency
  entry 1
    match-value nava
    caller-codec-list PCMU
callee-codec-list PCMA
    media police strip
    action cac-complete
    complete
  cac-policy-set global 1
  codec-list PCMU
  codec PCMU
  exit
  codec-list PCMA
  codec PCMA
  exit
end

Example: Configuring the Unified SBC to Enable Transrating

Transrating is supported only for different rates using the same codec, not across codecs. Therefore, transrating and transcoding cannot be performed simultaneously.

The following example shows how to configure the unified SBC for enabling the transrating feature using the same codec policy:

enable
configure terminal
sbc mySBC
sbe
  cac-policy-set 2
  first-cac-table Transrate
cac-table Transrate
  table-type policy-set
  entry 1
    cac-scope call
callee ptime 30
caller ptime 20
    media police strip
    action cac complete
    complete
  cac-policy-set global 2
end

The following example shows how to configure the Unified SBC for enabling the transrating feature using the same codec policy:

enable
configure terminal
sbc MySBC
sbe
  codec list PCMU
Example: Configuring the Unified SBC to Enable SRTP and Transcoding

The following example shows how to configure SBC to enable the SRTP and transcoding features.

```plaintext
enable
configure terminal
sbc mySBC
sbe
  cac-policy-set 3
  first-cac-table C3
cac-table c3
table-type policy-set
entry 1
cac-scope global
srtp support allow
srtp caller mandate
srtp callee mandate
srtp interworking allow
srtp media interworking allow
action next-table xcode
exit
exit
cac-table xcode
table-type limit dst-adjacency
entry 1
match-value nav4b
callee-codec-list PCMU
action cac-complete
complete
cac-policy-set global 3
end
```

Example: Configuring the Unified SBC for In-Band DTMF Interworking

The following example shows how to configure the unified SBC for inband DTMF transmission.

---

**Note**

The **caller** and **callee** commands have been used in this procedure. In some scenarios, the **branch** command can be used as an alternative to the **caller** and **callee** command pair. The **branch** command has been introduced in Release 3.5.0. See the "Configuring Directed Nonlimiting CAC Policies" section on page 7-37 for information about this command.

---

```plaintext
enable
configure terminal
sbc mySBC
sbe
  cac-policy-set 4
  first-cac-table c4
cac-table c4
table-type policy-set
entry 1
  cac-scope global
callee inband-dtmf-mode always
caller inband-dtmf-mode never
```
action next-table xcode
exit
exit
cac-table xcode
table-type limit dst-adjacency
entry 1
match-value spab
callee-codec-list PCMU
action cac-complete
complete
cac-policy-set global 4
end

Example: Configuring the Unified SBC to Support AMR-WB

The following example shows how to configure the Unified SBC to support AMR-WB:

enable
configure terminal
sbc mySBC
sbe
cac-policy-set 1
first-cac-scope dst-adjacency
first-cac-table codec-dst-acc
cac-table codec-dst-acc
table-type limit dst-adjacency
entry 1
match-value nava
caller-codec-list AMRWB
callee-codec-list PCMA
media police strip
action cac-complete
complete
cac-policy-set global 1
codec-list AMRWB
codec AMR-WB
exit
codec-list PCMA
codec PCMA
exit
Tracking Policy Failure Statistics

Users can track the number of calls that Cisco Unified Border Element (SP Edition) rejected based on the rules established in the number analysis policies, routing policies, or Call Admission Control (CAC) policies. Users can also view and query the policy failure statistics associated with these rejected calls, which can help them determine whether changes need to be made to the existing policies.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).

For a complete description of the commands used in this chapter, refer to the Cisco Unified Border Element (SP Edition) Command Reference: Unified Model at:


For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

For Cisco IOS XE Release 2.4, this feature is supported in the unified model only.

Feature History for Policy Failure Statistics

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 2.4</td>
<td>This feature was introduced on the Cisco IOS XR along with support for the unified model.</td>
</tr>
</tbody>
</table>

Contents

This module contains the following sections:

- Restrictions for Tracking Call Policy Failure Statistics, page 42-2
- Information About Policy Failure Statistics, page 42-2
Restrictions for Tracking Call Policy Failure Statistics

Review the following restrictions for policy failure statistics:

- Only new call failures are tracked by this feature.
- Only call failures associated with local policy are recorded. Calls rejected by downstream signaling devices are not included in this statistics.

Information About Policy Failure Statistics

The section provides information on the following:

- Policy Failure Statistics for a Specified Time Interval, page 42-2
- Policy Set and Per-Entry Statistics, page 42-2
- Automatic Tracking of Policy Failure Statistics, page 42-3
- Policy Failure Statistics and Hunting, page 42-4

Policy Failure Statistics for a Specified Time Interval

Table 42-1 lists the command to view and clear the failure statistics on the specified signaling border element (SBE) for a certain time interval.

<table>
<thead>
<tr>
<th>Table 42-1 Commands for Time-Based Policy Failure Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>clear sbc service-name sbe policy-failure-stats</td>
</tr>
<tr>
<td>Clears the policy failure statistics for the current and previous time interval.</td>
</tr>
</tbody>
</table>

Policy Set and Per-Entry Statistics

To determine whether calls failed due to policies configured in the routing, number validation, or CAC tables, users can view the policy failure statistics for a specific policy table or table entry. Table 42-2 lists the commands to view and clear the statistics on policy tables associated with a policy set.

<table>
<thead>
<tr>
<th>Table 42-2 Commands for Statistics for Policy Tables in a Policy Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>show sbc service-name sbe cac-policy-set policy set-id tables</td>
</tr>
<tr>
<td>Displays a summary of the CAC policy tables associated with the given policy set, including the number of failed calls.</td>
</tr>
<tr>
<td>clear sbc service-name sbe cac-rejection-stats</td>
</tr>
<tr>
<td>Clears all CAC policy failure statistics.</td>
</tr>
<tr>
<td>show sbc service-name sbe call-policy-set policy set-id tables</td>
</tr>
<tr>
<td>Displays a summary of routing policy tables associated with the given policy set, including the number of failed calls.</td>
</tr>
<tr>
<td>clear sbc service-name sbe call-rejection-stats</td>
</tr>
<tr>
<td>Clears all routing and number analysis policy rejection statistics.</td>
</tr>
<tr>
<td>show sbc service-name sbe cac-policy-set policy set-id table name entries</td>
</tr>
<tr>
<td>Displays the specified entire CAC policy table.</td>
</tr>
</tbody>
</table>
Table 42-3 lists the commands to view the detailed information for a specific entry in a CAC policy table and routing table.

Table 42-3  Per-Entry Statistics Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>show sbc service-name sbe cac-policy-set policy set-id table name entries</code></td>
<td>Displays a summary of the entries associated with the given CAC table.</td>
</tr>
<tr>
<td><code>show sbc service-name sbe call-policy-set policy set-id table name entry entry</code></td>
<td>Displays detailed statistics for the given entry in the routing table.</td>
</tr>
<tr>
<td><code>show sbc service-name sbe call-policy-set policy set-id table name entries</code></td>
<td>Displays a summary of the entries associated with the given routing table.</td>
</tr>
</tbody>
</table>

Automatic Tracking of Policy Failure Statistics

Cisco Unified Border Element (SP Edition) automatically tracks policy failure statistics for call attribute sets representing the following:

- Per source adjacency statistics for all configured adjacencies
- Per destination adjacency statistics for all configured adjacencies
- Per source account statistics for all configured accounts
- Per destination account statistics for all configured accounts

Table 42-4 lists the commands to view and clear automatically tracked policy failure statistics.

Table 42-4  Automatically Tracked Statistics Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>show sbc service-name sbe policy-failure-stats src-adjacency table-name period</code></td>
<td>Displays the policy statistics of the specified source adjacency for the specified time interval. The value for the <code>period</code> parameter must be one of the following:</td>
</tr>
<tr>
<td></td>
<td>- current5mins</td>
</tr>
<tr>
<td></td>
<td>- previous5mins</td>
</tr>
<tr>
<td></td>
<td>- current15mins</td>
</tr>
<tr>
<td></td>
<td>- previous15mins</td>
</tr>
<tr>
<td></td>
<td>- currenthour</td>
</tr>
<tr>
<td></td>
<td>- previoushour</td>
</tr>
<tr>
<td></td>
<td>- currentday</td>
</tr>
<tr>
<td></td>
<td>- previousday</td>
</tr>
<tr>
<td><code>clear sbc service-name sbe policy-failure-stats src-adjacency table-name</code></td>
<td>Clears the policy statistics of the specified source adjacency.</td>
</tr>
</tbody>
</table>

The statistics are collected at 5 minute intervals past the hour (that is, at 0, 5, 10, 15 minutes, and so on past the hour). For example, the periods covered by the various buckets at 12:43 would be as follows:

- current five minutes: 12:40-12:43
- previous five minutes: 12:35-12:40
Information About Policy Failure Statistics

- current 15 minutes: 12:30-12:43
- previous 15 minutes: 12:15-12:30
- current hour: 12:00-12:43
- last hour: 11:00-12:00
- current day: 00:00-12:43
- last day: 00:00-24h - 00:00

A counter can keep increasing. It keeps a count of events that have completed. When reporting the value of a counter, it's the sum total of events that happened in the period. Some examples of counters are total call attempts, failed call attempts, and active call failures.

A gauge is a counter that can go up and down. It typically tells you how many of something there are now. When reporting the value of a gauge, it's either the current value, or when measured over a longer time period, it's the average value during the period. Some examples of gauges are active calls, activating calls, and deactivating calls.

Policy Failure Statistics and Hunting

If the CAC module refuses a call or if a call cannot be signaled to the chosen destination adjacency because of a negative or no response, call hunting occurs. Call hunting is the process of selecting an alternative adjacency from the routing tables and retrying the call using the newly selected destination adjacency.

Hunting continues until one of the following conditions is fulfilled:
- The call gets connected.
- No further adjacencies are available for retry.
- The call has been hunted too many times.

Global Statistics and Call Hunting

If a call gets connected after hunting, Cisco Unified Border Element (SP Edition) does not include it in any of the following global statistics:
- Total call setup failures
- Total call setups failed due to number analysis
- Total call setups failed due to routing
- Total call setups failed due to CAC
- CAC failure due to number of calls limit
- CAC failure due to call rate limit
- CAC failure due to media channels limit
- CAC failure due to bandwidth limit

If a call fails after number analysis, hunting does not occur. Cisco Unified Border Element (SP Edition) includes it in the following global statistics:
- Total call setup failures
- Total call setups failed due to number analysis
If a call fails the first time it is routed because no destination adjacency is found in the routing table, then Cisco Unified Border Element (SP Edition) includes it in the following global statistics:

- Total call setup failures
- Total call setups failed due to routing

If a call fails because a CAC policy refused it permission to proceed, Cisco Unified Border Element (SP Edition) includes the failure in the total call setups failed due to CAC statistics. Additionally, the call is included in one of the following statistics depending on the nature of the CAC limit:

- CAC failure due to number of calls limit
- CAC failure due to call rate limit
- CAC failure due to media channels limit
- CAC failure due to bandwidth limit

**Per-table and Per-entry Statistics and Call Hunting**

If a call undergoes \( N \) iterations of hunting, then it traverses the number analysis tables once, and the routing and the CAC tables \( N \) times. But the CAC tables can reject the call each time it traverses the CAC table. For each time the CAC table rejects the call, Cisco Unified Border Element (SP Edition) finds the table and entry that was responsible for setting the CAC limit, and increments the following:

- Number of calls refused by the CAC table
- Number of calls refused by the table entry

**Per-adjacency and Per-Account Statistics and Call Hunting**

If a call gets connected after hunting, Cisco Unified Border Element (SP Edition) does not include it in the following per-account or per-adjacency statistics:

- total call setup failures
- total call setups failed due to number analysis
- total call setups failed due to CAC
- CAC failures due to rate limit
- CAC failures due to media channels limit
- CAC failures due to bandwidth limit

If a call fails due to number analysis, then hunting does not occur and Cisco Unified Border Element (SP Edition) includes the call in the following per-account and per-source adjacency statistics:

- total call setup failures
- total call setups failed due to number analysis

If a call fails in the routing tables before hunting occurs, Cisco Unified Border Element (SP Edition) includes the call in the following per-source account and per-source-adjacency statistics:

- total call setup failures
- total call setups failed due to routing

A call included in the total call setup failures statistics is included in the per-source adjacency, per-destination-adjacency, per-source-account adjacency, and per-destination account statistics. Additionally, if the most recent hunting attempt failed because a CAC policy refused the call permission to proceed, Cisco Unified Border Element (SP Edition) includes the failure in the total call setups failed due to CAC.
due to CAC statistics in the per-source-adjacency, per-destination-adjacency, per-source-account, and per-destination-account statistics. The call is also included in one of the following statistics depending on the nature of the CAC limit depending on the nature of the CAC limit:

- CAC failure due to number of calls limit
- CAC failure due to call rate limit
- CAC failure due to media channels limit
- CAC failure due to bandwidth limit
Implementing SNMP

Simple Network Management Protocol (SNMP) for Cisco Unified Border Element (SP Edition) is defined in the following MIBs:

- CISCO-SESSION-BORDER-CONTROLLER-EVENT-MIB—Defines SNMP notifications and alarms that are generated by Cisco Unified Border Element (SP Edition). This MIB sends the notifications and traps that are generated by Cisco Unified Border Element (SP Edition) to the SNMP manager.
- CISCO-SESSION-BORDER-CONTROLLER-STATS-MIB—Defines the SNMP statistics information for Cisco Unified Border Element (SP Edition). The two types are call statistics and media statistics. The calls are categorized as Session Initiation Protocol (SIP) calls and H.323 calls; the media statistics refer to RTP.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).

For a complete description of the commands used in this chapter, refer to the Cisco Unified Border Element (SP Edition) Command Reference: Unified Model at:


For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

Feature History for Implementing SNMP

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 2.4</td>
<td>Implementing Simple Network Management Protocol (SNMP) was introduced on the Cisco IOS XR.</td>
</tr>
</tbody>
</table>

Contents

- Prerequisites for Implementing SNMP, page 43-2
- Information About Implementing SNMP, page 43-2
- Implementing SNMP for Cisco Unified Border Element (SP Edition), page 43-3
- Configuration Example for Implementing SNMP, page 43-5
Prerequisites for Implementing SNMP

The following prerequisites are required to implement SNMP for Cisco Unified Border Element (SP Edition):

- You must have sufficient user privileges to modify the running configuration of the router.
- You must have configured Cisco Unified Border Element (SP Edition).
- Before you can access the SNMP MIBs to perform SNMPv2 polling for the SBC MIB statistics or to configure the SNMP users and groups for SNMPv3 polling, you must configure the SNMP read-only community string by using the `snmp-server community` command in Cisco IOS XR. For more information about the `snmp-server community` command, see the Cisco IOS Network Management Command Reference.

Information About Implementing SNMP

This section describes how to implement SNMP for SBC:

- SNMP Notifications, page 43-2
- SNMP Statistics, page 43-3

SNMP Notifications

Table 43-1 lists the types of SNMP notifications.

Table 43-1 List of SNMP Notifications

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Alert Notification</td>
<td>Specifies that the media is received from an unexpected source.</td>
</tr>
<tr>
<td>Blacklist Notification</td>
<td>Adds or removes the source from the blacklist table.</td>
</tr>
<tr>
<td>Adjacency Status Notification</td>
<td>Attaches or detaches the adjacency from the SBC.</td>
</tr>
<tr>
<td>SLA Violation Notification</td>
<td>Specifies that the violations of the Service Level Agreement (SLA) are described in the policy tables. SLAs can include the maximum number of calls allowed, maximum call rate, maximum bandwidth, and so forth.</td>
</tr>
<tr>
<td>Radius Connection Status Notification</td>
<td>Specifies that the SBC connection to a RADIUS server is either lost or restored.</td>
</tr>
<tr>
<td>H.248 Controller Status</td>
<td>Specifies if an H.248 controlled entity, either a data border element (DBE) or remote transcoder, is connected or detached from the SBC. See other sections of the documentation for supported H.248 controlled entities.</td>
</tr>
</tbody>
</table>
SNMP Statistics

Table 43-2 lists the types of SNMP statistics.

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global call statistics</td>
<td>Represents the global call-related statistics such as call rates, media flows, signaling flows, and so forth. The <code>show sbc dbe media-stats</code> command displays output for the global call statistics.</td>
</tr>
<tr>
<td>Periodic statistics</td>
<td>Represents the information for the SBC call statistics for a particular time interval such as current 5 minutes, previous 5 minutes, current 15 minutes, previous 15 minutes, current hour, and previous hour. The <code>show sbc sbc call-stats</code> command displays output for the periodic statistics.</td>
</tr>
<tr>
<td>Per flow statistics</td>
<td>Represents the SBC media flow statistics. These media statistics are used for each of the current ongoing call flows. The <code>show sbc dbe media-flow-stats</code> command displays output for the per-flow statistics.</td>
</tr>
<tr>
<td>H.248 statistics</td>
<td>Represents the information for the H.248 call-related statistics when the H.248 controller is associated with SBC. The <code>show sbc dbe controllers</code> command displays output for the H.248 statistics.</td>
</tr>
</tbody>
</table>

Implementing SNMP for Cisco Unified Border Element (SP Edition)

This section describes how to implement SNMP for Cisco Unified Border Element (SP Edition):

- Configuring SNMP Notifications, page 43-3

Configuring SNMP Notifications

Perform this task to configure SNMP notifications for Cisco Unified Border Element (SP Edition).

**SUMMARY STEPS**

1. `configure terminal`
2. `snmp-server enable traps sbc {adj-status | blacklist | h248-ctrlr-status | qos-statistics | radius-conn-status | sla-violation | source-alert}`
3. `end`
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters the global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> snmp-server enable traps sbc {adj-status</td>
<td>blacklist</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# snmp-server enable traps sbc blacklist</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> end</td>
<td>Exits the configuration command.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# end</td>
<td></td>
</tr>
</tbody>
</table>

- Use the **adj-status** keyword to enable the SNMP SBC Adjacency Status trap when an adjacency is attached to or detached from the SBE.
- Use the **blacklist** keyword to enable the SNMP SBC Dynamic Blacklist trap when a source is added or removed from the blacklist table.
- Use the **congestion-alarm** keyword to enable the SNMP SBC Congestion Alarm trap.
- Use the **h248-ctrlr-status** keyword to enable the SNMP SBC H.248 Controller Status trap. For a distributed deployment model, a DBE is attached or detached from the SBC.
- Use the **media-source** keyword to enable the SNMP SBC Media Source Alert traps.
- Use the **qos-statistics** keyword to enable the QoS statistics traps. See the **Implementing QoS Demarcation? section on page 38-10** for more information about the procedure.
- Use the **radius-conn-status** keyword to enable the SNMP SBC Radius Connection Status trap when the connection is changed for the RADIUS server.
- Use the **sla-violation** keyword to enable the SNMP SBC SLA Violation trap when there is an SLA violation in the policy tables. SLAs include the maximum number of calls allowed, maximum call rate, maximum bandwidth, and so on.
- Use the **source-alert** keyword to enable the SNMP SBC Source Alert trap when media is received from an unexpected source.

See the **Implementing SNMP Alarm Logs? section on page 44-6** for information about configuring logging for some of these alarms.
Configuration Example for Implementing SNMP

This section provides the following configuration example for implementing SNMP for Cisco Unified Border Element (SP Edition):

- Configuring SNMP Notifications: Example, page 43-5

Configuring SNMP Notifications: Example

The following example shows how to configure the SNMP blacklist notification for Cisco Unified Border Element (SP Edition):

```
configure terminal
snmp-server enable traps sbc blacklist
end
```
Logging Support

Cisco Unified Border Element (SP Edition) provides various features for working with logs. Logging can be configured so that logs are generated under specified conditions. Logs can also be generated on demand. Information derived from the logs can be used for analyzing and troubleshooting issues pertaining to the operation of the network and for identifying areas for improvement in the network.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).

For a complete description of the commands used in this chapter, refer to Cisco Unified Border Element (SP Edition) Command Reference: Unified Model at:

For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

Feature History for Logging Support

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 2.x</td>
<td>The Syslog feature was introduced in a release earlier than Cisco IOS XE</td>
</tr>
<tr>
<td></td>
<td>Release 3.1S.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 3.5S</td>
<td>The Call Log Correlation feature was introduced to enable all the correlation</td>
</tr>
<tr>
<td></td>
<td>logs associated with a particular call to be linked together using a</td>
</tr>
<tr>
<td></td>
<td>correlator ID.</td>
</tr>
<tr>
<td></td>
<td>The Alarms feature was enhanced to include new features for working with</td>
</tr>
<tr>
<td></td>
<td>alarm logs.</td>
</tr>
</tbody>
</table>

Contents

This chapter contains the following sections:

- Syslog Capabilities, page 44-2
- Call Log Correlation, page 44-4
- Alarm Logs, page 44-6
Syslog Capabilities

All the Cisco Unified Border Element (SP Edition) debug messages that are displayed on the console are recorded in the Cisco IOS syslog. All the Cisco IOS syslog commands that configure log size, persistence, and redirection can be used for managing the syslog.

In addition to the console messages, Cisco Unified Border Element (SP Edition) records a log in its own internal buffer. This is known as the problem determination log and is saved in the event of a software-forced reload or as a result of using the `sbc dump-diagnostics` command. When you compile the problem reports, the problem determination log file is included as part of the problem reports.

Internal Log Levels

The Session Border Controller (SBC) application uses an internal log level to control the verbosity of the console and the PD log. Although both the console and problem determination log levels can be changed independently, we do not recommend changing the problem determination log level because the problem determination log buffer is of limited size and important logs may be lost.

The default SBC problem determination logging level is 63 for the console and 60 for the buffer. You can change the default SBC problem determination logging level using the `debug sbc log-level console` command, the `debug sbc log-level filter` command, or the `debug sbc log-level buffer` command.

<table>
<thead>
<tr>
<th>Log Level</th>
<th>Syslog Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>90</td>
<td>Fatal</td>
</tr>
<tr>
<td>80</td>
<td>Error</td>
</tr>
<tr>
<td>70</td>
<td>Unexpected</td>
</tr>
<tr>
<td>63</td>
<td>Configuration Error</td>
</tr>
<tr>
<td>60</td>
<td>Operational</td>
</tr>
<tr>
<td>50</td>
<td>Audit</td>
</tr>
<tr>
<td>40</td>
<td>Statistics</td>
</tr>
<tr>
<td>30</td>
<td>Verbose Operational</td>
</tr>
<tr>
<td>20</td>
<td>Verbose Statistics</td>
</tr>
<tr>
<td>10</td>
<td>Internal Diagnostic</td>
</tr>
</tbody>
</table>

Enabling the Syslog Functionality

To enable the syslog functionality on the SBC, set the internal log levels, and issue the syslog-specific logging commands. The following example assumes a default problem determination level of 63 (no further action is needed if this is a fresh reboot).

1. Enable logging using the following commands:

   ```
   Router# configure
   Router(config)# logging enable
   Router(config)# logging standby
   ```
Note  The `logging standby` command allows the synchronization of the active and standby syslog settings.

2. Configure the location to which you want the syslog messages to be sent. Locations can be one of the following:
   - Console: `logging console <1-7>`
     
     ```
     Router(config)# logging console severity-level
     ```
   - Buffer: `logging buffer <1-7>`
     
     ```
     Router(config)# logging buffered severity-level
     ```

Note  Use the `show logging` command to view the logging statistics and the logging buffer. Use the `clear logging` command to clear the logging buffer.

   - Syslog server: `logging trap <1-7>`
     
     ```
     Router(config)# logging host ip_address [tcp[/port] | udp[/port]]
     Router(config)# logging trap severity-level
     Router(config)# logging device-id {hostname | ipaddress interface_name | string text | context-name}
     Router(config)# logging facility number
     ```

Note  The `logging device-id` command allows the customization of syslog messages when sending the log to a remote server.

   - Telnet sessions: `logging monitor <1-7>`
     
     ```
     Router(config)# logging monitor severity-level
     Router# terminal monitor
     ```

   - SNMP management station: `logging history <1-7>`
     
     ```
     Router(config)# logging history severity-level
     ```

   - Supervisor: `logging supervisor <1-7>`
     
     ```
     Router(config)# logging supervisor severity-level
     ```
3. Configure specific syslog message manipulation:

   Router(config)# logging message syslog_id [level severity_level]
   Router# show logging message
   Router# clear logging

4. Configure the global syslog settings:

   Router(config)# logging queue queue-size
   Router# show logging queue
   Router(config)# logging timestamp
   Router(config)# logging rate-limit {num (interval | level severity_level | message syslog_id) | unlimited (level severity_level | message syslog_id)}
   Router# show logging

Call Log Correlation

The Call Log Correlation feature enables all the correlation logs associated with a particular call to be linked together using a correlator ID. This feature also enables real-time filtering of logs on a particular call. A 64-bit diagnostics correlator is assigned to each SIP call, REGISTER, SUBSCRIBE, or NOTIFY messages.

You can set the filters based on the following parameters:

- Dialed or dialing number
- Session Initiation Protocol (SIP) Universal Resource Identifier (URI)
- Remote signaling address
- Remote VPN ID
- Adjacency
- VRF

The logs that match the selected filter type are saved in a separate problem determination trace file and inter process signal (IPS) trace file.

Use the following command to enable the correlation-logs filter:

   debug sbc sbc-name correlation-logs filter filter-name [pdtrc-log-level value]

Use the following command to disable the correlation-logs filter:

   no debug sbc sbc-name correlation-logs filter filter-name

Use the following command to display the debug logs, filters, and log levels:

   show debugging
## Problem Determination Log Levels

You can set the problem determination log level in the filter using the `pdtrc-log-level` option in the `debug sbc sbc-name correlation-logs filter filter-name [pdtrc-log-level value]` command. The problem determination trace log level ranges from 0 to 100. The default log level is 60. A log level of 100 indicates that no logs are output, and 0 indicates that all the logs are output.

Table 44-1 lists the problem determination log levels:

<table>
<thead>
<tr>
<th>Problem Determination Log Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>90</td>
<td>Critical system errors</td>
</tr>
<tr>
<td>80</td>
<td>Major system errors</td>
</tr>
<tr>
<td>70</td>
<td>Minor system errors</td>
</tr>
<tr>
<td>63</td>
<td>Configuration errors</td>
</tr>
<tr>
<td>60</td>
<td>Call errors</td>
</tr>
<tr>
<td>55</td>
<td>Call overview</td>
</tr>
<tr>
<td>50</td>
<td>Call details</td>
</tr>
<tr>
<td>40</td>
<td>Call statistics</td>
</tr>
<tr>
<td>30</td>
<td>Verbose operational</td>
</tr>
<tr>
<td>20</td>
<td>Verbose statistics</td>
</tr>
<tr>
<td>10</td>
<td>Internal diagnostic</td>
</tr>
</tbody>
</table>

## Examples of Call Log Correlation Feature

The following example shows the various filters available for filtering the correlation logs:

```
Router# debug sbc test correlation-logs filter ?
  adjacency                  Adjacency, matching calls to or from this adjacency
  dn                        Dialed/dialing number, matching calls to or from this number
  remote-signalling-address Remote signalling address matching to or from this address
  sip-uri                   SIP-URI, matching calls to or from this URI
  vrf                       VRF name
```

The following example shows the filtering of correlation logs based on the adjacency parameter:

```
Router# debug sbc test correlation-logs filter adjacency abc
Debugging filter log-level set to default level 60

Router# show debugging
  SBC correlator filter Adjacency name is abc
  IpsTracing is enabled
```

The following example shows the filtering of correlation logs based on the dialing number parameter:

```
Router# debug sbc test correlation-logs filter dn aa
Debugging filter log-level set to default level 60

Router# show debugging
```
SBC correlator filter DN is aa
IpsTracing is enabled

The following example shows the filtering of correlation logs based on the remote signalling address parameter:

Router# debug sbc test correlation-logs filter remote-signalling-address ipv4 192.0.2.1
Debugging filter log-level set to default level 60

Router# show debugging
SBC buffer log-level is 0
SBC correlator
Filter Remote signalling-address ipv4 address is 192.0.2.1
IpsTracing is enabled
SBC correlator
Filter DN is abc
Pd loglevel is 70
IpsTracing is enabled

The following example shows the filtering of correlation logs based on the SIP URI parameter:

Router# debug sbc test correlation-logs filter sip-uri ccc
Debugging filter log-level set to default level 60

Router# show debugging
SBC correlator filter Adjacency name is abc
IpsTracing is enabled
SBC correlator filter Remote signalling-address ipv4 address is 192.0.2.1
IpsTracing is enabled
SBC correlator filter SIP-URI is ccc
IpsTracing is enabled
SBC correlator filter DN is aa
IpsTracing is enabled

The following example shows the filtering of correlation logs based on the VRF parameter:

Router# debug sbc test correlation-logs filter vrf new ipv4 rsa 192.0.2.1 pdtrc-log-level 70
Debugging filter log-level set to default level 60

Router# show debugging
SBC correlator Filter Remote signalling-address ipv4 address is 192.0.2.1
SBC correlator Filter VRF is new with Vpn(id) = 3
Pd loglevel is 70
IpsTracing is enabled
SBC correlator Filter SIP-URI is 9.0.0.0
Pd loglevel is 0
IpsTracing is enabled

Alarm Logs

You can configure the SBC to generate alarms for various types of events associated with the operation of the SBC. You can also configure the SBC to log debugging information, which you can use to monitor and tune the functioning of the system. On the basis of the alarms, you can take corrective and preventive action to ensure that the SBC continues functioning according to your business requirements. It is also
Chapter 44      Logging Support

Alarm Logs

important to monitor the alarms generated by the SBC over a period of time and analyze this information. To address this requirement, you can configure the SBC to generate, display, and store alarm logs. The information provided in the alarm logs can help resolve some common issues, such as interoperability problems and incorrect configurations. These logs can also be used to identify issues that might potentially require escalation and investigation by more specialized support staff. Information in the logs can be used to improve the overall efficiency of the system.

Note

All alarm log information is lost after a route processor failover.

You can use any combination of the following commands to configure alarm logs:

- Use the `debug sbc alarm-filter` command to specify the alarm types for which alarm logs must be generated.
- Use the `debug sbc alarm-log-level` command to specify the output mode and the alarm severity level for which alarms must be logged.
- The buffer that is used to store alarm logs may run out of free space while log files are written to it. In addition, you may want to store alarm logs for future reference. Use the `sbc periodic-dump-alarms` command to configure periodic movement of alarm log files from the buffer to a file system.
- Use the `sbc dump-alarms` command to move the alarm logs from the buffer to either a file system that you specify or the default file system configured on the router.

Configuring Alarm Logs

This task explains the commands that you can use to configure alarm logs. Note that it is not mandatory to use any particular command described in this task. You can use any combination of these commands to configure alarm logs.

SUMMARY STEPS

1. `debug sbc sbc-name alarm-filter alarm-type`
2. `debug sbc sbc-name alarm-log-level [buffer | console] severity-level`
3. `sbc periodic-dump-alarms [dump-location file-system [time-period time-period] | time-period time-period]`
4. `sbc dump-alarms [file-system]`
5. `show debugging`
## Alarm Logs

### Detailed Steps

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Configures the alarm types for which alarm logs must be generated.</td>
</tr>
<tr>
<td><code>debug sbc sbc-name alarm-filter alarm-type</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong> Router# debug sbc MySbc alarm-filter audit-congestion</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Configures the output mode and the alarm severity level for which alarms must be logged.</td>
</tr>
<tr>
<td>`debug sbc sbc-name alarm-log-level [buffer</td>
<td>console] severity-level`</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# debug sbc MySbc alarm-log-level console 40</td>
<td></td>
</tr>
</tbody>
</table>

**Note**  
The size of a single log file created on the file system cannot exceed 2 MB. When the size of a particular log file reaches 2 MB, a new file is created and logging output is stored in the new file.

- **sbc-name**—Name of the SBC.
- **alarm-type**—One of the following alarms:
  - `audit-congestion`—Call audit congestion.
  - `blacklist-alert`—Blacklist alert.
  - `blacklist-event`—Blacklist event.
  - `h248`—H248 connection failed.
  - `handled-exception`—Handed exception.
  - `routing-component`—Routing component set not active.
  - `routing-config`—Routing config set not active.
  - `routing-invalid`—Invalid routing configuration.
  - `sip-congestion`—SIP congestion detection.
  - `sip-peer`—SIP peer unavailable.
  - `vqm`—Voice Quality metrics (VQM) threshold exceeded.
- **buffer**—Specifies that alarm logs must be stored in the buffer.
- **console**—Specifies that logging output must be displayed on the console.
- **severity-level**—Alarm severity level for which logs must be generated. The range is from 0 to 100. For alarm logs stored in the buffer, the default is 40. For alarm logs displayed on the console, the default is 80. To disable logging, set the value to 100. If you set the value to 0, logs are generated for all levels of alarm severity.
## Alarm Logs

### Step 3

```plaintext
sbc periodic-dump-alarms {dump-location file-system [time-period time-period] | time-period time-period}
```

**Example:**

```plaintext
Router(config-sbc)# sbc periodic-dump-alarms dump-location bootflash: time-period 120
```

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>sbc periodic-dump-alarms</strong></td>
<td>Configures periodic movement of alarm log files from the buffer to a file system.</td>
</tr>
<tr>
<td><strong>dump-location</strong></td>
<td>Specifies that you want the alarm logs to be stored in a file system.</td>
</tr>
<tr>
<td><strong>file-system</strong></td>
<td>Name of the file system where you want the alarm logs to be moved. For example, <code>file-system</code> can be one of the following:</td>
</tr>
<tr>
<td>- <code>bootflash</code>:</td>
<td></td>
</tr>
<tr>
<td>- <code>flash</code>:</td>
<td></td>
</tr>
<tr>
<td>- <code>fpd</code>:</td>
<td></td>
</tr>
<tr>
<td>- <code>ftp</code>:</td>
<td></td>
</tr>
<tr>
<td>- <code>http</code>:</td>
<td></td>
</tr>
<tr>
<td>- <code>https</code>:</td>
<td></td>
</tr>
<tr>
<td>- <code>obfl</code>:</td>
<td></td>
</tr>
<tr>
<td>- <code>pram</code>:</td>
<td></td>
</tr>
<tr>
<td>- <code>rcp</code>:</td>
<td></td>
</tr>
<tr>
<td>- <code>scp</code>:</td>
<td></td>
</tr>
<tr>
<td>- <code>tftp</code>:</td>
<td></td>
</tr>
<tr>
<td><strong>time-period</strong></td>
<td>Specifies the periodic time interval, in minutes, after you want the logs to be moved. The range is from 0 to 1440. The default is 60.</td>
</tr>
</tbody>
</table>

**Note** When you run the **no** form of this command, the time period for moving logs is set to 0 and periodic movement of the logs is disabled.
Chapter 44  Logging Support

Alarm Logs

The following sample output of the `show debugging` command shows the debugging settings created by running the `debug sbc alarm-filter` command and the `debug sbc alarm-log-level` command. In this example, these debug commands have been used to specify that logs must be generated for call audit congestion alarms that are of severity level 60 or higher and that these logs must be moved to the specified file system at 120-minute intervals:

```
Router# show debugging

SBC:
  SBC buffer alarm-log-level : 60
  SBC alarm filter 1 : AUDIT CONGESTION
  SBC alarm periodic dump time : 120 min
```

### Step 4

**Command or Action**

```
sbc dump-alarms [file-system]
```

**Example:**

```
Router{config-sbc-sbe-cacpolicy)# sbc dump-alarms bootflash:
```

**Purpose**

Moves alarm logs from the buffer to either a file system that you specify or the default file system configured on the router.

- `file-system`—Name of the file system where you want the alarm logs to be moved. For example, `file-system` can be one of the following:
  - `bootflash`
  - `flash`
  - `fpd`
  - `ftp`
  - `http`
  - `https`
  - `obfl`
  - `pram`
  - `rcp`
  - `scp`
  - `tftp`

### Step 5

**Command or Action**

```
show debugging
```

**Example:**

```
Router# show debugging
```

**Purpose**

Displays information about the types of debugging that are enabled on the router.

The output of this command includes debugging settings created by running the `debug sbc alarm-filter` command and the `debug sbc alarm-log-level` command.
SIP 3xx Redirect Responses

This section describes how Cisco Unified Border Element (SP Edition) can be configured to process Session Initiation Protocol (SIP) 3xx responses. 3xx is a class of the response code used in SIP to indicate that further action needs to be taken in order to complete the request. The sender of the request should retry the request, using one or more alternative Uniform Resource Identifiers (URIs), which are presented in the 3xx response.

For Cisco IOS XE Release 2.4, this feature is supported in the unified model only.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).

For a complete description of the commands used in this chapter, refer to the Cisco Unified Border Element (SP Edition) Command Reference: Unified Model at:


For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

Feature History for SIP 3xx Redirect Responses

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 2.4</td>
<td>This feature was introduced on the Cisco IOS XR along with support for the unified model.</td>
</tr>
</tbody>
</table>

Contents

This module contains the following sections:

- Information About 3xx Redirect Responses in SIP, page 45-2
- How to Configure Cisco Unified Border Element (SP Edition) to Process SIP 3xx Responses, page 45-3
- Examples of Configuring Cisco Unified Border Element (SP Edition) to Process SIP 3xx Responses, page 45-5
Information About 3xx Redirect Responses in SIP

This section contains the following subsections:

- 3xx Responses, page 45-2
- Diversion Headers, page 45-3

3xx Responses

3xx responses are usually only expected in session-initiating requests, INVITEs. However, the SIP specification does not preclude sending 3xx responses for other request types. A number of alternative URIs are supplied on the 3xx responses in Contact headers.

The 3xx class of responses includes any response code in the range of 300-399 and indicates a redirection of the call. The redirection requires further action to be taken to complete the request. The following 3xx response codes are defined in SIP:

- **300 Multiple Choices.** The address in the request resolved to several choices, each with its own specific location. The user or user agent (UA) can select a preferred communication end point and redirect the request to that location.
  - The response may include a message body containing a list of resource characteristics and location(s), from which the user or UA can choose the most appropriate one, if allowed by the Accept request header field. However, no MIME types have been defined for this message body.
  - The choices should also be listed as Contact fields. The response may contain several Contact fields or a list of addresses in a Contact field. UAs may use the Contact header field value for automatic redirection or ask the user to confirm a choice.

- **301 Moved Permanently.** The user can no longer be found at the address in the Request-URI, and the requesting client should retry at the new address given by the Contact header field. The requestor should update any local directories, address books, and user location caches with this new value, and redirect future requests to the addresses listed.

- **302 Moved Temporarily.** The requesting client should retry the request at the new address(es) given by the Contact header field. The Request-URI of the new request uses the value of the Contact header field in the response.
  - The duration of the validity of the Contact URI can be specified through an Expires header field or an Expires parameter in the Contact header field. Both proxies and UAs may cache this URI for the duration of the expiration time. If there is no explicit expiration time, the address is valid only once for recursing, and must not be cached for future transactions.
  - If the URI cached from the Contact header field fails, the Request-URI from the redirected request may be tried again only once.

- **305 Use Proxy.** The requested resource must be accessed through the proxy given by the Contact field. The Contact field gives the URI of the proxy. The recipient is expected to repeat this single request via the proxy. 305 responses must only be generated by the user agent servers (UASs).

- **380 Alternative Service.** The call was not successful, but alternative services are possible. The alternative services are described in the message body of the response. There are no formats currently defined for this information.
  - In each case, the request should be retried to one of the supplied alternative URIs. The request can be retried by either the originating UA, or by an intermediate back-to-back user agent (B2BUA) or proxy on behalf of the originating UA (and without notifying it).
Cisco Unified Border Element (SP Edition) is a B2BUA, and, therefore, in some deployments it may be necessary for Cisco Unified Border Element (SP Edition) to retry the request instead of sending a negative response back to the initiator of the request.

**Diversion Headers**

The Diversion header enables the called SIP user agent to identify from whom the call was diverted and why it was diverted. The header notifies the original caller:

- That the call has been redirected to a destination that differs from the original target
- The number to which the call has been redirected
- The reason for the redirection

The diversion header is attached by networking elements that change the final destination of a request.

**How to Configure Cisco Unified Border Element (SP Edition) to Process SIP 3xx Responses**

This section contains the steps for configuring Cisco Unified Border Element (SP Edition) to process SIP 3xx responses.

**Configuring Cisco Unified Border Element (SP Edition) to Process SIP 3xx Responses**

**SUMMARY STEPS**

1. `configure terminal`
2. `sbc sbc-name`
3. `sbe`
4. `adjacency sip adjacency-name`
5. `redirect-mode mode`
6. `attach`
7. `exit`
8. `redirect-limit limit`
9. `end`
10. `show sbc sbc-name sbe adjacencies`
11. `show sbc sbc-name sbe redirect-limit`
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc service-name</td>
<td>Enters the mode of an SBC service. Use the service-name argument to define the name of the service.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# sbc mysbc</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Enters the mode of the signaling border element (SBE) function of the SBC.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> adjacency sip adjacency-name</td>
<td>Enters the mode of an SBE SIP adjacency. Use the adjacency-name argument to define the name of the service.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe)# adjacency sip SipToIsp42</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> redirect-mode mode</td>
<td>Configures the behavior of the SBC on receipt of a 3xx response to an INVITE from the SIP adjacency.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-adj-sip) redirect-mode recurse</td>
<td>- redirect-mode pass-through—SBC passes all 3xx responses back to the caller (the default mode).</td>
</tr>
<tr>
<td></td>
<td>- redirect-mode recurse—On 300, 301, 302, and 305 INVITE responses (under the redirect-limit, see Step 8), SBC resends the INVITE to the first listed contact address. Otherwise, SBC passes 3xx responses back.</td>
</tr>
<tr>
<td></td>
<td>- no redirect-mode—The no version of this command returns the adjacency to the default behavior.</td>
</tr>
<tr>
<td><strong>Step 6</strong> attach</td>
<td>Attaches the adjacency.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-adj-sip)# attach</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> exit</td>
<td>Exits the adjacency-sip mode and returns to the SBE mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-adj-sip)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> redirect-limit limit</td>
<td>Configures the maximum number of redirections that the SBC performs on a given call.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe)# redirect-limit 4</td>
<td>- redirect-limit limit—A numeric value, the maximum number of redirections performed before the call is failed (the range is 0-100, the default is 2).</td>
</tr>
<tr>
<td></td>
<td>- no redirect-limit—The no version of this command returns the adjacency to the default behavior.</td>
</tr>
</tbody>
</table>
This section provides two simple configurations for processing SIP 3xx responses.

The following command configures the adjacency “SipToIsp42” to recurse on 300, 301, 302, and 305 INVITE responses.

```
Router(config)# sbc mySbc sbe adjacency sip SipToIsp42
Router(config-sbc-sbe-adj-sip)# redirect-mode recurse
Router(config-sbc-sbe-adj-sip)# end
```

The following command configures the SBE to perform maximum 4 SIP 3xx redirections per call.

```
Router(config)# sbc mySbc
Router(config-sbc)# sbe
Router(config-sbc-sbe)# redirect-limit 4
Router(config-sbc-sbe)# end
```
SIP Call Hold

The Session Initiation Protocol (SIP) call hold feature in Cisco Unified Border Element (SP Edition) provides a standard telephony service of putting a caller on hold. If a party in a call wants to put the other party on hold, a party re-invites the other by sending an INVITE request with a modified Session Description Protocol (SDP). When a SIP endpoint wishes to place a call on hold or respond to a call hold re-INVITE, it chooses an appropriate method. Cisco Unified Border Element (SP Edition) modifies call hold SDPs to use any available methods in order to maximize inter-operating with SIP devices.

For Cisco IOS XE Release 2.4, this feature is supported in the unified model only.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).

For a complete description of the commands used in this chapter, refer to the Cisco Unified Border Element (SP Edition) Command Reference: Unified Model at:

For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

Feature History for SIP Call Hold

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 2.4</td>
<td>This feature was introduced on the Cisco IOS XR along with support for the unified model.</td>
</tr>
</tbody>
</table>

Contents

This module contains the following sections:

- Information About SIP Call Hold in Cisco Unified Border Element (SP Edition), page 46-2
- How to Configure SIP Call Hold, page 46-2
- SDP Call Hold Interworking, page 46-3
- Configuration Examples, page 46-10
Information About SIP Call Hold in Cisco Unified Border Element (SP Edition)

Cisco Unified Border Element (SP Edition) accepts a SIP re-INVITE with an SDP, signaling that the sender wishes to put the call on hold. Cisco Unified Border Element (SP Edition) modifies the SDP offer as needed and replaces remote endpoint addresses with known data border element (DBE) media addresses. Cisco Unified Border Element (SP Edition) then forwards the SIP message, containing the modified SDP to the remote endpoint.

If the re-INVITE is rejected by the endpoint going on hold, then the error response is returned to the holding endpoint (the endpoint that initiated the call hold). The media gate on the DBE continues to be connected and media continues to flow as before.

How to Configure SIP Call Hold

This section contains the steps for configuring the “no media” timeout duration for on-hold calls.

Configuring SIP Call Hold

SUMMARY STEPS

1. configure terminal
2. sbc service-name
3. sbe
4. hold-media-timeout timeout
5. end
6. show sbc service-name sbe hold-media-timeout
7. show sbc service-name sbe calls

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>Purpose</strong></td>
</tr>
<tr>
<td><strong>configure</strong></td>
<td>Enables global configuration mode.</td>
</tr>
</tbody>
</table>
| **Example:**
  Router# configure                  |                                                                         |
| **Step 2**                         | **Purpose**                                                             |
| **sbc service-name**               | Enters the mode of an SBC service.                                      |
| **Example:**
  Router(config)# sbc mysbc         | Use the service-name argument to define the name of the service.        |
SDP Call Hold Interworking

Cisco IOS XE Release 2.4 introduces support for SDP call hold interworking. With SDP call hold interworking, there are two ways of setting up call hold using SIP. Either the caller or callee can renegotiate the call characteristics using SDP so that either:

- The connection line is set to the null address, c=IN IP4 0.0.0.0.
- Or to the direction attribute for their endpoint so that it does not receive media from the endpoint.
  - If this was previously set to a=sendrecv, the endpoint putting the call on hold sets it to a=sendonly.

### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>sbe</td>
<td>Enters the mode of the signaling border element (SBE) function of the SBC.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>hold-media-timeout timeout</td>
<td>The time an SBE will wait after receiving a media timeout notification from the DBE for an on hold call before tearing that call down.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe)# hold-media-timeout 7200</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>end</td>
<td>Exits the configuration session and enters Privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe)# end</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>show sbc sbc-name sbe hold-media-timeout</td>
<td>Shows the currently configured duration of the media timeout timer for on-hold calls.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router# show sbc mysbc sbe hold-media-timeout</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>show sbc sbc-name sbe calls</td>
<td>Lists all the calls on the SBE.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router# show sbc mysbc sbe calls</td>
<td></td>
</tr>
</tbody>
</table>
– If this was previously set to a=recvonly, the endpoint putting the call on hold sets it to a=inactive.

Some SIP endpoints support setting the connection line to the null address, some support setting the direction, and some support both approaches. Additionally, some endpoints only respect setting the direction attribute to sendonly or inactive.

With SDP call hold interworking, Cisco Unified Border Element (SP Edition) supports interoperating with SIP endpoints that support a subset of the above approaches. When Cisco Unified Border Element (SP Edition) detects that a call is being put on hold in the SDP, it removes any preexisting c=IN IP4 0.0.0.0 or a=direction lines and replaces them with appropriate settings for the endpoint.

If the endpoint putting the call on hold was sendrecv or sendonly, then the default behavior is to send
- c=IN IP4 0.0.0.0
- a=sendonly

If the endpoint putting the call on hold was recvonly or inactive, then the default behavior is to send
- C=IN IP4 0.0.0.0
- a=inactive

**Note**
Music on hold is supported using SDP call hold interworking.

### Configuring SDP Call Hold Interworking

**SUMMARY STEPS**

1. configure terminal
2. sbc service-name
3. sbe
4. cac-policy-set policy-set-id
5. first-cac-scope scope-name
6. first-cac-table table-name
7. cac-table table-name
8. table-type {policy-set | limit \{list of limit tables\}}
9. entry entry-id
10. cac-scope {list of scope options}
11. match-value key
12. caller-hold-setting {hold-c0 | hold-c0-inactive | hold-c0-sendonly | hold-sendonly | standard}
13. action [cac-complete | next-table goto-table-name ]
14. exit
15. exit
16. complete
17. active-cac-policy-set policy-set-id
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>configure terminal</code></td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td><code>sbc service-name</code></td>
<td>Enters the mode of an SBC service. Use the <code>service-name</code> argument to define the name of the service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config)# sbc mysbc</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td><code>sbe</code></td>
<td>Enters the mode of the signaling border element (SBE) function of the SBC.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td><code>cac-policy-set policy-set-id</code></td>
<td>Enters the submode of CAC policy set configuration within an SBE entity.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe)# cac-policy-set 1</td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td><code>first-cac-scope scope-name</code></td>
<td>Configures the scope at which to begin defining limits when performing the admission control stage of policy. The <code>scope-name</code> argument configures the scope at which limits should be initially defined. Possible values are: adj-group, call, category, dst-account, dst-adj-group, dst-adjacency, dst-number, global, src-account, src-adj-group, src-adjacency, src-number. Features can be enabled or disabled per adjacency group through CAC configuration the same way this is done per individual adjacencies.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-cacpolicy)# first-cac-scope global</td>
<td></td>
</tr>
</tbody>
</table>
## SDP Call Hold Interworking

### Chapter 46      SIP Call Hold

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 6</strong></td>
<td></td>
</tr>
<tr>
<td><code>first-cac-table table-name</code></td>
<td>Configures the name of the first policy table to process when performing the admission control stage of policy.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-sbc-sbe-cacpolicy)# first-cac-table</code></td>
<td>RootCacTable</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td></td>
</tr>
<tr>
<td><code>cac-table table-name</code></td>
<td>Enters the mode for configuration of an admission control table (creating one if necessary) within the context of an SBE policy set.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router(config-sbc-sbe-cacpolicy)# cac-table</code></td>
<td>RootCacTable</td>
</tr>
</tbody>
</table>
### Step 8

**Command or Action**

```
step 8
```

**table-type {policy-set | limit} (list of limit tables))

**Example:**

```
Router(config-sbc-sbe-cacpolicy-cactable)#
table-type limit event-type
```

Configures the table type of a CAC table within the context of an SBE policy set.

*list of limit tables* can be one of the following values:

- **account**—Compare the name of the account.
- **adj-group**—Compare the name of the adjacency group.
- **adjacency**—Compare the name of the adjacency.
- **all**—No comparison type. All events match this type.
- **call-priority**—Compare with call priority.
- **category**—Compare the number analysis assigned category.
- **dst-account**—Compare the name of the destination account.
- **dst-adj-group**—Compare the name of the destination adjacency group.
- **dst-adjacency**—Compare the name of the destination adjacency.
- **dst-prefix**—Compare the beginning of the dialed digit string.
- **event-type**—Compare with CAC policy event types.
- **src-account**—Compare the name of the source account.
- **src-adj-group**—Compare the name of the source adjacency group.
- **src-adjacency**—Compare the name of the source adjacency.
- **src-prefix**—Compare the beginning of the calling number string.

Features can be enabled or disabled per adjacency group through CAC configuration the same way this is done per individual adjacencies. The adj-group table type matches on either source or destination adjacency group.

### Step 9

**Command or Action**

```
entry entry-id
```

**Example:**

```
Router(config-sbc-sbe-cacpolicy-cactable)# entry
```

Creates or modifies an entry in a table.
### Command or Action

**Step 10**

`cac-scope {list of scope options}`

### Purpose

Configures the scope within each of the entries at which limits are applied in a policy set table.

- **list of scope options**—Specifies one of the following strings used to match events:
  
  - **account**—Events that are from the same account.
  
  - **adjacency**—Events that are from the same adjacency.
  
  - **adj-group**—Events that are from members of the same adjacency group.
  
  - **call**—Scope limits are per single call.
  
  - **category**—Events that have same category.
  
  - **dst-account**—Events that are sent to the same account.
  
  - **dst-adj-group**—Events that are sent to the same adjacency group.
  
  - **dst-adjacency**—Events that are sent to the same adjacency.
  
  - **dst-number**—Events that have same destination.
  
  - **global**—Scope limits are global
  
  - **src-account**—Events that are from the same account.
  
  - **src-adj-group**—Events that are from the same adjacency group.
  
  - **src-adjacency**—Events that are from the same adjacency.
  
  - **src-number**—Events that have the same source number.
  
  - **sub-category**—The limits specified in this scope apply to all events sent to or received from members of the same subscriber category.
  
  - **sub-category-pfx**—The limits specified in this scope apply to all events sent to or received from members of the same subscriber category prefix.
  
  - **subscriber**—The limits specified in this scope apply to all events sent to or received from individual subscribers (a device that is registered with a Registrar server).

**Example:**

```
Router(config-sbc-sbe-cacpolicy-cactable-entry)#
cac-scope call
```
### SDP Call Hold Interworking

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 11** match-value key | Specifies the keyword used to match events. The format of the key is determined by the table-type. If you configure either an event-type or call-priority Limit table, then you only see the keyword options that apply for that type of Limit table. For Limit event-type tables (table-type limit event-type), the match value keyword options are the following:  
- call-update—Compare the beginning of the calling number string.  
- endpoint-reg—Compare the name of the destination adjacency.  
- new-call—Compare the beginning of the dialed digit string.  
For Limit call-priority tables (table-type limit call-priority), the match value keyword options are the following:  
- critical—Match calls with resource priority ‘critical.’  
- flash—Match calls with resource priority ‘flash’.  
- flash-override—Match calls with resource priority ‘flash-override.’  
- immediate—Match calls with resource priority ‘immediate.’  
- priority—Match calls with resource priority ‘priority.’  
- routine—Match calls with resource priority ‘routine.’  
For all other Limit tables, enter a name or digit string.  
WORD—Name or digit string to match. (Max Size 255). |
| **Example:** Router(config-sbc-sbe-cacpolicy-cactable-entry)# match-value call-update | |

| **Step 12** caller-hold-setting (hold-c0 | hold-c0-inactive | hold-c0-sendonly | hold-sendonly | standard) | Configures the caller hold settings that are supported. |
| Example: Router(config-sbc-sbe-cacpolicy-cactable-entry)# caller-hold-setting hold-sendonly | |

| **Step 13** action [cac-complete | next-table goto-table-name] | Specifies the action to take if this routing entry is chosen. |
| Example: Router(config-sbc-sbe-cacpolicy-cactable-entry)# action cac-complete | |
Configuration Examples

This section contains configuration examples.

Example of Configuring SIP Call Hold

The following command configures the SBE to wait for two hours after receiving the last media packet on an on-hold call before cleaning up the call resources.

```
Router# configure
Router(config)# sbc mysbc
Router(config-sbc)# sbe
Router(config-sbc-sbe)# hold-media-timeout 7200
```

Example of Configuring SDP Call Hold Interworking

In the example below, Fairchild Foods have replaced all the phones in their offices. The new phones support setting a=sendonly and c=IN IP4 0.0.0.0 to place a call on hold; they do not support setting a=inactive. You now want to reconfigure Cisco Unified Border Element (SP Edition) to work with these phones without changing the behavior for other customers. This change creates new policies at the account scope for all events, so that calls in which Fairchild Foods phones are involved are put on hold appropriately.

The following configuration changes will make sure Fairchild phone doesn’t receive a=inactive in SDP when Fairchild is the source account and the callee puts the call on hold.

```
Router# configure terminal
Router(config)# sbc mysbc
Router(config-sbc)# sbe
Router(config-sbc-sbe)# cac-policy-set 1
```

---

### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td><code>exit</code></td>
<td>Exits the cactable entry configuration mode and enters the cactable mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-cacpolicy-cactable-entry)# exit</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td><code>exit</code></td>
<td>Exits the cactable configuration mode and enters the cacpolicy mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-cacpolicy-cactable)# exit</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td><code>complete</code></td>
<td>Completes the CAC-policy or call-policy set after committing the full set.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-cacpolicy)# complete</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td><code>active-cac-policy-set policy-set-id</code></td>
<td>Sets the active CAC-policy-set within an SBE entity.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router (config-sbc-sbe)# active-cac-policy-set 1</td>
<td></td>
</tr>
</tbody>
</table>

---

OL-19820-15

Router(config-sbc-sbe-cacpolicy)# first-cac-scope global
Router(config-sbc-sbe-cacpolicy)# first-cac-table callhold-src-settings
Router(config-sbc-sbe-cacpolicy)# cac-table callhold-src-settings
Router(config-sbc-sbe-cacpolicy-cactable)# table-type limit src-account
Router(config-sbc-sbe-cacpolicy-cactable)# entry 1
Router(config-sbc-sbe-cacpolicy-cactable-entry)# match-value fairchild
Router(config-sbc-sbe-cacpolicy-cactable-entry)# caller-hold-setting hold-c0-sendonly
Router(config-sbc-sbe-cacpolicy-cactable-entry)# action cac-complete
Router(config-sbc-sbe-cacpolicy-cactable-entry)# exit
Router(config-sbc-sbe-cacpolicy-cactable)# exit
Router(config-sbc-sbe-cacpolicy)# complete

The following configuration changes will make sure Fairchild phone doesn't receive a=inactive in SDP when Fairchild is the destination account and the caller puts the call on hold.

Router# configure terminal
Router(config)# sbc mysbc
Router(config-sbc)# sbe
Router(config-sbc-sbe)# cac-policy-set 1
Router(config-sbc-sbe-cacpolicy)# first-cac-scope global
Router(config-sbc-sbe-cacpolicy)# first-cac-table callhold-dst-settings
Router(config-sbc-sbe-cacpolicy)# cac-table callhold-dst-settings
Router(config-sbc-sbe-cacpolicy-cactable)# table-type limit dst-account
Router(config-sbc-sbe-cacpolicy-cactable)# entry 1
Router(config-sbc-sbe-cacpolicy-cactable-entry)# match-value fairchild
Router(config-sbc-sbe-cacpolicy-cactable-entry)# callee-hold-setting hold-c0-sendonly
Router(config-sbc-sbe-cacpolicy-cactable-entry)# action cac-complete
Router(config-sbc-sbe-cacpolicy-cactable-entry)# exit
Router(config-sbc-sbe-cacpolicy-cactable)# exit
Router(config-sbc-sbe-cacpolicy)# complete
SIP Call Transfer

Cisco Unified Border Element (SP Edition) supports Session Initiation Protocol (SIP) call transfer, a standard Internet telephony service. Call transfer allows a wide variety of decentralized multiparty call operations. These decentralized call operations form the basis for third-party call control, and are important features for voice over IP (VoIP) and SIP. Call transfer is also critical for conference calling, where calls can transition smoothly between multiple point-to-point links and IP level multicasting. The Cisco Unified Border Element (SP Edition) SIP call transfer feature includes basic in-dialog transfer and advanced call transfer for the following network topologies:

- Central SBC
- Transfer intra network
- Transfer out of network
- Transfer to colleague

For Cisco IOS XE Release 2.4, this feature is supported in the unified model only.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).

For a complete description of the commands used in this chapter, refer to the Cisco Unified Border Element (SP Edition) Command Reference: Unified Model at:


For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

Feature History for SIP Call Transfer

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 2.4</td>
<td>This feature was introduced on the Cisco IOS XR along with support for the unified model.</td>
</tr>
</tbody>
</table>
Contents

This module contains the following sections:

- Restrictions for SIP Call Transfer Support, page 47-2
- Information About SIP Call Transfer, page 47-2

Restrictions for SIP Call Transfer Support

The following is a list of restrictions for SIP call transfer support:

- The Configuration feature is expected to be “always on.” Therefore, no configuration is required and it is not possible to disable it.
- REFER subscription state is not maintained over failover. Therefore, after a failover, any subsequent NOTIFYs telling the one referring about the progress of the referral are lost. They are bounced back with a 481 SIP error response. This will not prevent calls from being transferred, but may result in a few error logs if diagnostics are enabled.

Information About SIP Call Transfer

REFER Requests

The REFER method has three main roles:

- Originator—User agent that initiates the transfer or REFER request.
- Recipient—User agent that receives the REFER request and is transferred to the final-recipient.
- Final-Recipient—User agent introduced into a call with the recipient.

The REFER method always begins within the context of an existing call and starts with the originator. The originator sends a REFER request to the recipient (user agent receiving the REFER request) to initiate a triggered INVITE request. The triggered INVITE request uses the SIP URL contained in the Refer-To header as the destination of the INVITE request.

The recipient then contacts the resource in the Refer-To header (final-recipient), and returns a SIP 202 (Accepted) response to the originator. The recipient also must notify the originator of the outcome of the REFER transaction—whether the final-recipient was successfully or unsuccessfully contacted. The notification is accomplished using the Notify Method, SIP's event notification mechanism.

A Notify message with a message body of SIP 200 OK indicates a successful transfer, while a body of SIP 503 Service Unavailable indicates an unsuccessful transfer. If the call was successful, a call between the recipient and the final-recipient results.

Cisco Unified Border Element (SP Edition) accepts and passes through in-dialog REFER requests. Standard SIP headers are manipulated as normal. The call-transfer specific headers are treated in the following way:

- The Refer-To header is passed through unchanged.
- The Referred-By header:
  - Any received Referred-By header is passed through ignored.
On the outbound REFER, the following header is written:

```
Referred-By: <sip:endpoint_dn@sbc_adj_sip_domain_name>
```

except that,

If the side of the call on which Cisco Unified Border Element (SP Edition) received the REFER has privacy enabled (configured in CAC), then no Referred-By header is written on the outbound REFER.

- The Replaces header is treated in the same way as for the INVITE requests.

Out-of-dialog REFER requests are rejected. The Target-Dialog header is not explicitly supported, and therefore is stripped or passed through, subject to header and method white/blacklisting configuration.

### NOTIFY Messages

When the outcome of the REFER transaction is known, the recipient of the REFER request must notify the originator of the outcome of the REFER transaction—whether the final-recipient was successfully or unsuccessfully contacted. The notification is accomplished using the NOTIFY method, SIP's event notification mechanism. The notification contains a message body with a SIP response status line and the response class in the status line indicates the success or failure of the REFER transaction.

Cisco Unified Border Element (SP Edition) accepts and passes through in-dialog NOTIFY requests. Standard SIP headers are manipulated as normal.

- If the NOTIFY contains a body of type message/sipfrag, and if the start of this body can be correctly parsed as a SIP response status line, then the outbound NOTIFY is given a message/sipfrag body containing a SIP response status line with the same response code (and nothing else).
- If there is no body of type message/sipfrag on the NOTIFY, or the first line of the NOTIFY body cannot be correctly parsed as a status line, then the onbound NOTIFY is sent without a body. This includes the case where there is a message/sipfrag body included as part of a mime/multipart body.

### Replaces Headers

The processing of Replaces headers is the key logic involved in supporting call transfer across Cisco Unified Border Element (SP Edition). Cisco Unified Border Element (SP Edition) does a lookup on the call IDs and tags in the received Replaces header. If it finds the corresponding call branch (for example, C1), then it looks up the partner call branch (for example, C2). C1 and C2 together make up another call through Cisco Unified Border Element (SP Edition). The Replaces header sent out on the request which is forwarded on might refer to call branch C1 or C2, depending on the request type and other considerations. Any “early-only” flag on the Replaces header is passed through.
SIP Authentication


Note
For Cisco IOS XE Release 2.4, this feature is supported in the unified model only.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).

For a complete description of the commands used in this chapter, refer to the Cisco Unified Border Element (SP Edition) Command Reference: Unified Model at:


For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

Feature History for SIP Authentication

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 2.4</td>
<td>Support for SIP authentication was introduced on the Cisco ASR 1000 Series Aggregation Services Routers.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 2.5</td>
<td>Support for interoperability for SIP authentication of INVITE requests was introduced on the Cisco ASR 1000 Series Aggregation Services Routers.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 2.6</td>
<td>Support for interoperability for SIP authentication of outbound out-of-dialogue requests (using the same generation scheme as used by INVITE requests for the Call-ID, From and To dialog tags, and CSeq sequence numbers) was introduced on the Cisco ASR 1000 Series Aggregation Services Routers.</td>
</tr>
</tbody>
</table>

Contents

This module contains the following sections:

- SIP Outbound Authentication, page 48-2
- SIP Inbound Authentication, page 48-6
- Interoperability for SIP Authentication, page 48-11
SIP Outbound Authentication

When network entities communicate using SIP, one entity often needs to challenge another one to determine if it is authorized to transmit SIP signaling into the challenger’s network. The SIP authentication model is based on the HTTP digest authentication, as described in the RFC 2617. The use of basic authentication, where passwords are transmitted unencrypted, is not permitted in SIP.

This section contains the following subsections:

- Prerequisites for Implementing SIP Outbound Authentication, page 48-2
- Restrictions for Implementing SIP Outbound Authentication, page 48-2
- Information About SIP Outbound Authentication, page 48-3
- How to Configure SIP Outbound Authentication, page 48-4
- Examples of Show Commands, page 48-5

Prerequisites for Implementing SIP Outbound Authentication

The following prerequisites are required to implement SIP outbound authentication:

- Configure a SIP adjacency before you specify one or more authentication-realms.
- Configure the Cisco Unified Border Element (SP Edition) with a set of domains (realms) with which it can authenticate itself. Set the username and password to provide when challenged by each of these domains. This configuration is implemented per adjacency.

**Note**

Multiple realms can be configured per adjacency and there is no limit on the number of these realms aside from memory availability. Different realms may be configured with the same username and password. Also, each realm may be configured with different username and password on different adjacencies. However, any realm can be configured a maximum of one time per adjacency.

Restrictions for Implementing SIP Outbound Authentication

The following restrictions apply to SIP outbound authentication:

- Cisco Unified Border Element (SP Edition) rejects any attempt to configure an authentication-realm with the same domain name as an existing authentication-realm. This restriction is valid per adjacency. Multiple adjacencies may have authentication-realms configured with the same domain.

**Note**

The current command line interface (CLI) prohibits the user from configuring two authentication-realms with the same domain for the same adjacency. If this is attempted, the CLI interprets the second authentication-realm configuration as an attempt to reconfigure the first authentication-realm, and updates the user’s credentials accordingly.

- Each authentication-realm can only be configured with a single username and password per adjacency.
Information About SIP Outbound Authentication

This section contains the following subsections:

- Configuring Outbound Authentication in Cisco Unified Border Element (SP Edition), page 48-3
- Authenticating the Cisco Unified Border Element (SP Edition) to Remote Devices, page 48-3

Configuring Outbound Authentication in Cisco Unified Border Element (SP Edition)

When a SIP adjacency is configured, the user may specify one or more authentication-realms. Each authentication-realm represents a remote domain, from which Cisco Unified Border Element (SP Edition) receives authentication challenges on the adjacency. When an authentication-realm is configured, the user must specify the correct user name and password that Cisco Unified Border Element (SP Edition) uses to authenticate itself in that realm. Cisco Unified Border Element (SP Edition) stores all valid authentication-realms for each adjacency.

Authenticating the Cisco Unified Border Element (SP Edition) to Remote Devices

Upon receipt of a SIP 401 or 407 response that can be correlated to a request it sent, Cisco Unified Border Element (SP Edition) examines the attached authentication challenge. Cisco Unified Border Element (SP Edition) responds to any authentication challenge received on a given adjacency that matches one of the configured authentication-realms for that adjacency. Any authentication challenge that does not match the configured authentication-realm is passed through unchanged to the SBC’s signaling peer for the adjacency, on which the original request was received.

To generate a response to an authentication challenge, Cisco Unified Border Element (SP Edition) does the following:

1. First, it looks up the realm parameter of the challenge in its list of configured authentication-realms for the outbound adjacency.
2. Second, it finds the password for that authentication-realm and generates an authentication response by combining the password with the nonce parameter from the challenge, and hashing the result.
3. If the challenger has requested auth-int quality of protection, Cisco Unified Border Element (SP Edition) also generates a hash of the entire message body and includes it in the response.
4. Cisco Unified Border Element (SP Edition) builds an Authorization (or Proxy-Authorization) header by including the following parameter values (following RFC 2617):
   - Nonce from challenge.
   - Realm from challenge.
   - Digest-URI is set to the SIP URI of the challenged request.
   - Message-QOP is set to auth.
   - Response calculated as described previously.
   - Username as specified for the relevant authentication-realm.
   - If the challenge contained an opaque parameter, it is returned unchanged on the response.
   - If the challenge contained the qop-directive parameter, then the nonce-count parameter is set to the number of the sent requests, using the response calculated from this nonce.
Note that the domain parameter is not expected to be included on any challenges that Cisco Unified Border Element (SP Edition) must respond to. This parameter is not used on Proxy-Authenticate challenges, the type of challenge that Cisco Unified Border Element (SP Edition) most often receives. If the domain parameter is included, Cisco Unified Border Element (SP Edition) ignores it.

Finally, Cisco Unified Border Element (SP Edition) stores its calculated response and the received nonce with the other data for the authentication-realm. This allows Cisco Unified Border Element (SP Edition) to respond rapidly to the subsequent challenges from this realm with the same nonce. If Cisco Unified Border Element (SP Edition) lacks the resources to store its response, it carries on anyway. The next time an authorization challenge is received from this realm, Cisco Unified Border Element (SP Edition) has to re-calculate its response. When Cisco Unified Border Element (SP Edition) re-uses a saved response, it updates the nonce count stored along with the nonce-response pair. This allows Cisco Unified Border Element (SP Edition) to correctly fill in the nonce-count field in Authorization responses.

How to Configure SIP Outbound Authentication

This section contains the steps for configuring SIP outbound authentication, allowing the user to add/remove one or more authentication-realms to/from an adjacency.

**SUMMARY STEPS**

1. configure terminal
2. sbc service-name
3. sbe
4. adjacency sip adjacency-name
5. authentication-realm inbound domain | outbound domain username password
6. end
7. show sbc sbc-name sbe adjacency adjacency-name authentication-realms
8. show sbc service-name sbe all-authentication-realms

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure terminal</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router# configure terminal</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>sbc service-name</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config)# sbc mysbc</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>sbe</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc)# sbe</td>
</tr>
</tbody>
</table>
## Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| Step 4 | `adjacency sip adjacency-name` | Enters the mode of an SBE SIP adjacency.  
  - Use the `adjacency-name` argument to define the name of the service. |
| Example: | `Router(config-sbc-sbe)# adjacency sip test` | |
| Step 5 | `authentication-realm {inbound domain|outbound domain username password}` | Configures a set of outbound authentication credentials for the specified domain on the specified adjacency. This command can be issued either before or after the adjacency has been attached.  
  The no version of this command deconfigures the authentication-realm on the specified adjacency.  
  - inbound—Specifies inbound authentication realm.  
  - outbound—Specifies outbound authentication realm.  
  - domain—Name of the domain for which the authentication credentials are valid.  
  - username—User name that identifies the SBC in the specified domain.  
  - password—Password to authenticate the username in the specified domain. |
| Example: | `Router(config-sbc-sbe-adj-sip)# authentication-realm outbound example.com usersbc passwordsbc` | |
| Step 6 | `end` | Exits the adj-sip mode and returns to privileged EXEC mode |
| Example: | `Router(config-sbc-sbe-adj-sip)# end` | |
| Step 7 | `show sbc sbc-name sbe adjacency adjacency-name authentication-realms` | Shows all currently configured authentication-realms for the specified SIP adjacency. |
| Example: | `Router# show sbc mySbc sbe adjacency SipToIsp42 authentication-realms` | |
| Step 8 | `show sbc service-name sbe all-authentication-realms` | Shows all currently configured authentication-realms for all SIP adjacencies. |
| Example: | `Router# show sbc mySbc sbe all-authentication-realms` | |

## Examples of Show Commands

```
Router# show sbc mySbc sbe adjacency SipToIsp42 authentication-realms

Configured authentication realms
-----------------------------------------------
Domain   Username    Password
Example.com usersbc passwordsbc
```

```
Router# show sbc mySbc sbe all-authentication-realms
```
Configured authentication realms
--------------------------------

Adjacency: SipToIsp42
Domain       Username  Password     Example.com  usersbc   passwordsbc
Remote.com   usersbc   sbcpassword

Adjacency: SipToIsp50
Domain       Username  Password
Example.com  user2sbc  password2sbc
Other.com    sbcuser   sbcsbcbsbc

SIP Inbound Authentication

Cisco Unified Border Element (SP Edition) supports two modes of Session Initiation Protocol (SIP) inbound authentication to challenge inbound SIP requests: local and remote. You must select the mode of authentication to configure Cisco Unified Border Element (SP Edition) according to the level of support present in the Remote Authentication Dial-In User Service (RADIUS) servers. If the RADIUS servers are compliant with only draft-sterman-aaa-sip-00 to 01, then select the local mode. If the RADIUS servers are compliant with only RFC 4590, then use the remote authentication mode.

Note
This feature is optional and you can configure the Cisco Unified Border Element (SP Edition) not to challenge the inbound requests.

This section contains the following subsections:

- Prerequisites for Implementing SIP Inbound Authentication, page 48-6
- Restrictions for Implementing SIP Outbound Authentication, page 48-2
- Information About SIP Inbound Authentication, page 48-7
- How to Configure SIP Inbound Authentication, page 48-8
- Examples of Show Commands, page 48-5

Prerequisites for Implementing SIP Inbound Authentication

The following prerequisites are required to implement SIP inbound authentication:

- Configure a SIP adjacency with the intended mode of authentication before you configure Cisco Unified Border Element (SP Edition) to authenticate inbound calls.
- Configure the RADIUS server to specify which mode of inbound authentication is selected.

Restrictions for Implementing SIP Inbound Authentication

The following restrictions and limitations apply to implement SIP inbound authentication:

- Cisco Unified Border Element (SP Edition) supports only one inbound authentication realm per adjacency.
- Cisco Unified Border Element (SP Edition) does not check the validity of nonces generated by a RADIUS server; the RADIUS server must be configured to perform this check.
Cisco Unified Border Element (SP Edition) does not designate a particular RADIUS server group on an adjacency for inbound authentication.

Since trust-transference of calls does not occur between inbound authentication, outbound authentication, and Transport Layer Security (TLS) connections, a successful inbound authentication does not ensure that Cisco Unified Border Element (SP Edition) marks the call as secure or implement outbound authentication. Users can, however, configure inbound authentication, outbound authentication, and TLS independently on the same adjacency.

### Information About SIP Inbound Authentication

This section contains the following subsections:

- Local Inbound Authentication, page 48-7
- Remote Inbound Authentication, page 48-7
- Interaction with Outbound Authentication, page 48-7
- Failure Modes for Inbound Authentication, page 48-7

### Local Inbound Authentication

When configured to perform local inbound authentication, Cisco Unified Border Element (SP Edition) is responsible for challenging an unauthorized request from the remote peer first. Therefore, to be able to challenge the request from the remote peer, the adjacency must already be configured with an authentication realm. After the remote peer has validated the request, it is forwarded to the RADIUS server, which then decides whether to permit the call to pass through or not.

### Remote Inbound Authentication

When configured to perform remote inbound authentication, Cisco Unified Border Element (SP Edition) relies on the RADIUS server to challenge an authorized request from the remote peer. Cisco Unified Border Element (SP Edition) forwards the challenge request generated by the RADIUS server to the remote peer, and also forwards the remote peer’s authentication request to the RADIUS server.

### Interaction with Outbound Authentication

If an adjacency is configured for inbound authentication, then after it successfully authenticates an inbound request, the authorization headers matching the realm for that adjacency are stripped out and not propagated to the outbound signal. Authorization headers for other realms, however, are passed through to the outbound request.

### Failure Modes for Inbound Authentication

When the inbound authentication is configured, the following failure modes may occur (in addition to the standard SIP signal failure modes):
Unacceptable Parameters

If the endpoint or RADIUS server specifies a quality of protection parameter other than **auth** or **auth-int**, then the inbound request is rejected and a 403 response is generated. Similarly, Cisco Unified Border Element (SP Edition) generates a 403 response when algorithms other than MD5 and MD5-sess are used.

Access-Request Rejection

If the RADIUS server rejects the Access-Request signal with an Access-Reject response, Cisco Unified Border Element (SP Edition) sends a 403 response to the endpoint.

Insufficient Memory

If Cisco Unified Border Element (SP Edition) does not have sufficient memory to process an inbound authentication request, it rejects the request and sends a 503 response.

No Match on Authentication Realm

If the peer does not return any authentication headers that specify the authentication realm contained in the adjacency’s configuration, then Cisco Unified Border Element (SP Edition) rechallenges the request with 401 response.

No Match on Nonce

If the peer’s nonce does not match the one generated by Cisco Unified Border Element (SP Edition), then Cisco Unified Border Element (SP Edition) rejects the authentication request and sends a 403 response.

Nonce Timed Out

If the peer’s nonce has timed out, then Cisco Unified Border Element (SP Edition) challenges the nonce by sending a 401 response and a new nonce.

No Acceptable RADIUS Servers

If there is no RADIUS server to support a mode configured on the adjacency, then Cisco Unified Border Element (SP Edition) rejects the authentication request with a 501 response and creates a log to alert the user of the inconsistent configuration.

How to Configure SIP Inbound Authentication

This section contains the steps for configuring SIP local inbound authentication a RADIUS server.

**SUMMARY STEPS**

1. `configure terminal`
2. `sbc service-name`
3. `sbe`
4. `radius [accounting client-name | authentication]`
5. `server server-name`
## Chapter 48      SIP Authentication

### SIP Inbound Authentication

6. address
7. mode local
8. key password
9. exit
10. activate
11. exit
12. adjacency sip adjacency-name
13. authentication-realm inbound realm
14. authentication mode local
15. authentication nonce timeout time
16. exit

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>configure terminal</strong></td>
</tr>
<tr>
<td>Example:</td>
<td>Router# configure terminal</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><strong>sbc service-name</strong></td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config)# sbc mysbc</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><strong>sbe</strong></td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc)# sbe</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>**radius [accounting client-name</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe)# radius authentication</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td><strong>server server-name</strong></td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-auth)# server authserv</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td><strong>address ipv4 ipv4-address</strong></td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-auth-ser)# address ipv4 200.200.200.122</td>
</tr>
</tbody>
</table>
### Step 7
```
mode {local|remote}
or
server server-name mode {local|remote}
```
Configure the RADIUS server for local inbound authentication. By default, the mode is remote.

**Example:**
```
Router(config-sbc-sbe-auth-ser)# mode local
```

### Step 8
```
key password
```
Sets the authentication server key.

**Example:**
```
Router(config-sbc-sbe-auth-ser)# key authpass1
```

### Step 9
```
exit
```
Exits the mode for configuring the authentication server.

**Example:**
```
Router(config-sbc-sbe-auth-ser)# exit
```

### Step 10
```
activate
```
Activates the RADIUS client.

**Example:**
```
Router(config-sbc-sbe-auth)# activate
```

### Step 11
```
exit
```
Exits the mode for configuring the RADIUS client and enters the SBE mode.

**Example:**
```
Router(config-sbc-sbe-auth)# exit
```

### Step 12
```
adjacency sip adjacency-name
```
Enters the mode of an SBE SIP adjacency.

- Use the `adjacency-name` argument to define the name of the service.

**Example:**
```
Router(config-sbc-sbe)# adjacency sip test
```

### Step 13
```
authentication-realm inbound realm
```
Configures a set of authentication credentials for a specified domain on the specified SIP adjacency.

**Note** This is a mandatory parameter for local mode.

**Example:**
```
Router(config-sbc-sbe-adj-sip)# authentication-realm inbound cisco.com
```

### Step 14
```
authentication mode local
```
Configures the SIP adjacency for local inbound authentication. To configure the SIP adjacency, for remote inbound authentication, set the value to `remote`.

**Example:**
```
Router(config-sbc-sbe-adj-sip)# authentication mode local
```

### Step 15
```
authentication nonce timeout time
```
Configures the value of the authentication nonce timeout in seconds. The range of acceptable values is 0 to 65535 seconds. The default value is 300 seconds.

**Example:**
```
Router(config-sbc-sbe-adj-sip)# authentication nonce timeout 10000
```

### Step 16
```
exit
```
Exits the adj-sip mode and returns to the SBE mode.

**Example:**
```
Router(config-sbc-sbe-adj-sip)# exit
```
Examples of Show Commands

Router# show sbc mySbc sbe adjacencies SipToIsp42 detail

SBC server mySbc
Adjacency SipToIsp42
Status: Attached
Signaling address: 10.2.0.122:5060
Signaling-peer: 200.200.200.179:8888
Force next hop: No
Account: core
Group: None
In Header Profile: Default
Out Header Profile: Default
In method profile: Default
Out method profile: Default
In UA option profile: Default
Out UA option profile: Default
In proxy option profile: Default
Priority set name: Default
Local-id: None
Rewrite REGISTER: Off
Target address: None
NAT Status: Auto-Detect
Reg-min-expiry: 3000 seconds
Fast-register: Enabled
Fast-register-int: 30 seconds
Authenticated mode: Local
Authenticated realm: Cisco.com
Authenticated nonce life time: 300 seconds
IMS visited NetID: None
Inherit profile: Default
Force next hop: No
Home network ID: None
UnEncrypt key data: None
SIPIpasssthrough: No
Rewrite from domain: Yes
Rewrite to header: Yes
Media passthrough: No
Preferred transport: UDP
Hunting Triggers: Global Triggers
Redirect mode: Passthrough

Interoperability for SIP Authentication

Cisco Unified Border Element (SP Edition) supports interoperability between SIP devices and third-party soft switch equipments for SIP authentication of all SIP requests. The supported interoperability applies to dialog-creating INVITE requests and out-of-dialog REGISTER and SUBSCRIBE requests only.

Support for interoperability for SIP authentication of INVITE requests was introduced in Cisco IOS XE Release 2.5. Cisco Unified Border Element (SP Edition) uses a generation scheme that generates the Call-ID, From and To dialog tags, and CSeq sequence numbers on the outbound call leg using the inbound request message data which provides both uniqueness and retains the same values for subsequent requests resulting from any challenges.

Support for interoperability for SIP authentication of out-of-dialog requests was introduced in Cisco IOS XE Release 2.6. The same generation scheme used by INVITE requests (based on the inbound request message data) was implemented for out-of-dialog requests.
Cisco Unified Border Element (SP Edition) interoperates with third-party soft switch devices for processing SIP authentication of INVITE and out-of-dialog requests in the following way:

- The Call-ID of an authorized SIP request matches that of the initial request.
- The To and From headers of an authorized SIP request match those of the initial request, including the dialog tag (if any) in the From header.
- The CSeq sequence number of an authorized SIP request is one higher than the initial request.

This section contains the following subsections:

- Information About SIP Outbound Authentication, page 48-3
- Information About Interoperability for SIP Authentication, page 48-13

Restrictions for Interoperability for SIP Authentication

The following restrictions apply to support for Interoperability for SIP Authentication on the Cisco Unified Border Element (SP Edition):

- Cisco Unified Border Element (SP Edition) meets the following interoperability requirements only when the received signaling from the upstream call leg also meets the same:
  - The Call-ID of an authorized SIP request matches that of the initial request.
  - The To and From headers of an authorized SIP request match those of the initial request, including the dialog tag (if any) in the From header.
  - The CSeq sequence number of an authorized SIP request is one higher than the initial request.
- Cisco Unified Border Element (SP Edition) depends on the randomness of the values in the received signaling. For example, if a calling endpoint generates insufficiently random values, then the values sent by the SBC on the outbound call leg will also be insufficiently random. However, the SBC uses any and all randomness from the Call-IDs and From tags generated by the caller, and in addition takes steps to avoid Call-ID and tag collisions between the upstream and downstream signaling.
- If the input from certain configuration fields to Call-ID and To/From header generation is changed between forwarding an initial SIP request and the subsequent authorized SIP request to that adjacency, then the headers of the two requests do not match. This can lead to call setup failure, depending on the downstream signaling entities. In particular, note the following:
  - The local-id or signaling-address configuration under adjacency affects Call-ID and From-tag generation.
  - Header rewriting configuration can apply to From and To headers. To meet the interoperability requirements, this rewriting must produce identical results on successive requests.
  - Cisco Unified Border Element (SP Edition) does not issue warnings to the user before accepting such configuration changes.
- Interoperability for SIP authentication affects dialog-creating INVITE requests and out-of-dialogue requests. Requests with other methods are not affected.
- To meet the interoperability requirements, the initial and authorized requests should not be routed out of different adjacencies.
Information About Interoperability for SIP Authentication

This section provides information about interoperability for SIP authentication.

SIP Requests

SIP requests refer to the messages within the scope of a single challenge or response sequence. There can be several sequences before a request is accepted, but the first sequence of any pair of consecutive requests is referred to as the initial request and the second one is referred to as the authorized request. SIP requests are both dialog-creating INVITE requests and out-of-dialog requests.

The following SIP request terms are used in this chapter:

- **Initial request**—A SIP request with insufficient authentication credentials, which is challenged with a “401–Unauthorized” or a “407–Proxy Authentication Required” response.
- **Authorized request**—The corresponding subsequent request, sent on receipt of the 401/407 challenge response. This request contains an extra Authorization or Proxy-Authorization header.

Call-ID Generation

Cisco Unified Border Element (SP Edition) generates Call-ID values for outbound dialog-creating INVITE requests and out-of-dialogue requests (such as REGISTERS and SUBSCRIBES), based on the Call-ID of the inbound request and on configuration.

The generated Call-ID values are composed of a 32-character hexadecimal MD5 hash of the received Call-ID, an ‘@’ character, and a local-id string representing the SBC itself.

Example:

```
Call-ID: 4264330abc5106c8ab70ed3fd222b7b2@sbc.home.net
```

Note

The local-id string is the configuration from the outbound adjacency. If this configuration is absent, the canonical text representation of the signaling-address from the outbound adjacency is used.

From Tag Generation

Cisco Unified Border Element (SP Edition) generates From header dialog tag values for outbound dialog-creating INVITE requests and out-of-dialogue requests (such as REGISTERS and SUBSCRIBES), based on the From tag of the inbound request and on configuration.

The generated From tag values are composed of a local-id string representing the SBC itself, two eight-character hexadecimal MD5 hashes of the received From tag, and a numerical index identifying the internal component responsible for the dialog.

Example:

```
From: "Fred" <sip:22222222@sbc.home.net>;tag=sbc.home.net+1+a27d9765+b7f0f7e1
```

The local-id string is generated from configuration in the same way as for Call-IDs.
Interoperability for SIP Authentication

Chapter 48  SIP Authentication

CSeq Sequence Number Generation

Cisco Unified Border Element (SP Edition) chooses the sequence number for use in the CSeq header of an outbound dialog-creating INVITE request and out-of-dialogue requests (such as REGISTERS and SUBSCRIBES) to be the same as the sequence number on the received inbound request.

Cisco Unified Border Element (SP Edition) continues to choose sequence numbers for subsequent outbound requests on the same dialog by storing the dialog’s current sequence number, and incrementing it each time a new transaction is created.

Example:
CSeq: 949005087 INVITE

Pass-Through Authentication

SBC supports passing through authentication challenges and their responses. No configuration is required for this.

- 407 responses are passed through by SBC.
- On challenge responses, a quality of protection (qop) of “auth-int” is stripped out, as SBC necessarily modifies the message, which negates the integrity of the authentication. If this is the only qop offered, the challenge is converted into a 403 Forbidden return code.
Late-to-Early Media Interworking

The late-to-early media interworking feature is supported for Session Initiation Protocol (SIP) calls. In order to interwork between a late media caller and an early media callee, Cisco Unified Border Element (SP Edition) sends an invite to the callee that includes a Session Description Protocol (SDP) offer of media. Two implementations of late-to-early media interworking are available:

- By default, SBC generates the SDP with a single media line that specifies codecs common to both the caller and the callee’s codec whitelists.
- SBC can also be configured with a media description using the `sip sdp-media-profile` command to generate a customized offer.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).

For a complete description of the commands used in this chapter, refer to the Cisco Unified Border Element (SP Edition) Command Reference: Unified Model at:


For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

Feature History for Late-to-Early Media Interworking

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 2.4</td>
<td>This feature was introduced on the Cisco IOS XR.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 2.5</td>
<td>The customizable offer for late-to-early media interworking feature was introduced on the Cisco IOS XR.</td>
</tr>
</tbody>
</table>

Contents

This module contains the following sections:

- Restrictions for Late-to-Early Media Interworking Support, page 49-2
- Information about Late-to-Early Media Interworking, page 49-2
- Configuring Late-to-Early Media Interworking, page 49-4
- Configuration Examples for the Late-to-Early Media Interworking Feature, page 49-13
- Verification, page 49-17
Restrictions for Late-to-Early Media Interworking Support

The restrictions for late-to-early media interworking are:

- This feature applies only to SIP-to-SIP calls, it does not apply to SIP-to-H.323 interworking calls.
- This feature applies only to IPv4; you cannot use it with IPv6 addressing.
- If the caller refines the media chosen by the callee, this is sent back to the callee in a PRACK. However, if the callee attempts to refine the media again, the event is logged but it is not passed back to the caller.
- Because Cisco Unified Border Element (SP Edition) generates SDPs, any calls using this feature cannot use media bypass.
- Cisco Unified Border Element (SP Edition) only generates SDPs offering a single audio stream. If the caller and callee want to negotiate video, fax, or other media streams, they can renegotiate this after the call has been established.
- If the callee attempts to send early media either before or without sending a reliable 1XX INVITE, Cisco Unified Border Element (SP Edition) will drop that media. It will not reach the caller.
- The callee must not send unreliable 1XX INVITE responses because the caller would interpret them as an out-of-sequence SDP offer. For late-to-early interworking calls, Cisco Unified Border Element (SP Edition) sets 100rel as mandatory in order to forbid the callee from sending unreliable responses only if the caller side supports 100rel.
- Late-to-early media interworking must not be used with the Gq IMS interface. This interface does not provide Cisco Unified Border Element (SP Edition) with the local media address necessary to create an SDP offer (and will likely result in calls with incorrect media paths).

Information about Late-to-Early Media Interworking

This section includes the following topics:

- Late-to-Early Media Interworking Description, page 49-2
- Customizable Offer for Late-to-Early Media Interworking, page 49-3

Late-to-Early Media Interworking Description

Early Media is the ability of two user agents to communicate before a call is actually established. Early Media can flow when the caller makes a media proposal on the initial call setup request and the callee responds to the offer before the call is connected. Cisco Unified Border Element (SP Edition) provides interoperability between SIP devices that do not provide SDP on their INVITEs and SIP devices that require SDP on INVITEs they receive. This occurs when:

- An endpoint caller wants to negotiate media after the INVITE has been accepted (late media) and does not include an SDP offer on the initial INVITE
- The callee that expects an SDP offer on the initial INVITE, which it then answers with a 1XX response (early media).

The normal negotiation for media is for the caller to include an SDP offer on the initial INVITE and for the callee to accept with a 200 response. However, the following might occur:

- Late media is used by some endpoints, such as call agents that want to allow the callee to select the media used.
Early media is used by some more recent endpoints that need to support media flow before the call is accepted, such as a pre-call announcement or in-band tones from a Call Hold server.

In order to interwork between a late media caller and an early media callee, Cisco Unified Border Element (SP Edition) sends an invite to the callee that includes an SDP offer of media. Cisco Unified Border Element (SP Edition) then sends appropriate messages between the caller and callee, depending on the responses from each.

Cisco Unified Border Element (SP Edition) supports this interworking on a per-adjacency basis. You can configure each adjacency to require late-to-early media interworking for calls made to that adjacency and/or for calls made from that adjacency.

### Customizable Offer for Late-to-Early Media Interworking

By default, SBC generates the SDP with a single media description that specifies codecs common to both the caller and callee’s codec whitelists.

The Customizable Offer for Late-to-Early Media Interworking feature provides customized SDPs with one or more media descriptions. You configure the media descriptions in named profiles (SDP media profiles) and associate the profiles to signals by including the profile name in a CAC policy.

To enable a customized offer for late-to-early media interworking:

- Enable late-to-early media interworking per adjacency, as described in the Configuring Late-to-Early Media Interworking Per Adjacency section on page 49-4.
- Create a named SDP media profile containing one or more media description lines which will be inserted into the SDP when SBC is generating the INVITE. SBC will insert the media description lines into the SDP per the sequence number configured.
- Associate this sdp-media-profile with a cac-policy table entry.

When a call requires late-to-early interworking, if the CAC policy entry for that call contains a valid SDP media profile name, then SBC generates a customized SDP. In the absence of such an association, SBC generates the default SDP. In the customized case, SBC inserts the media description lines in the media profile in the SDP when it generates the INVITE. Each entry in the media profile includes a sequence number, which controls the ordering of the lines in the generated SDP.

### Rules for Media Lines in SDP Media Profiles

A section of SDP is configured as an entry in the SDP Media profile. An entry can have one or many media description lines. The format of an SDP Media profile is:

```
entry_number
  media-line index "media_description"
  media-line index "media_description"
  exit
```

For example:

```
entry 1
  media-line 1 "m=audio 0 RTP/AVP 0"
  media-line 2 "a=rtpmap:0 PCMU/8000"
  exit
```

If more than one media description is created in the same profile, all of the entries are used to generate the same output SDP, in ascending order by entry number.
The media_description argument must be enclosed in quotes (" "). The value inside the quotes must be syntactically valid SDP as defined in RFC 2327. The following rules apply:

- An SDP entry must contain exactly one m-line. The m-line must appear first in the entry. The m-line port must be zero. SBC replaces the zero with the appropriate port.
- An SDP entry must not contain a c-line.

The Cisco command line interface handles the contents of media_description as a string value. It does not check the syntax of the configured information. If the syntax is incorrect, outbound offers by the SBC are rejected.

### Configuring Late-to-Early Media Interworking

This section describes the following configuration scenarios for Late-to-Early Media Interworking:

- Configuring Late-to-Early Media Interworking Per Adjacency, page 49-4
- Configuring Customized Offers for Late-to-Early Media Interworking, page 49-11

### Configuring Late-to-Early Media Interworking Per Adjacency

This task shows how to configure late-to-early media interworking per adjacency.

---

**Note**

The caller and callee commands have been used in this procedure. In some scenarios, the branch command can be used as an alternative to the caller and callee command pair. The branch command has been introduced in Release 3.5.0. See the Configuring Directed Nonlimiting CAC Policies? section on page 7-37 for information about this command.

---

**SUMMARY STEPS**

1. configure terminal
2. sbc service-name
3. sbe
4. adjacency sip adjacency-name
5. nat force-off
6. preferred-transport udp
7. redirect-mode pass-through
8. authentication nonce timeout value
9. signaling-address ipv4
10. signaling-port
11. remote-address ipv4
12. signaling-peer
13. signaling-peer-port
14. dbe-location-id
15. account
16. reg-min-expiry
17. media-late-to-early-iw { incoming | outgoing }
18. attach
19. exit
20. exit
21. sip inherit profile
22. cac-policy-set
23. first-cac-table
24. first-cac-scope
25. averaging-period
26. cac-table
27. table-type limit list of limit tables
28. entry
29. match-value
30. action cac-complete
31. max-bandwidth
32. max-updates
33. max-channels
34. early-media-type
35. early-media-timeout
36. codec-restrict-to-list
37. caller-codec-list
38. callee-privacy
39. caller-privacy
40. exit
41. exit
42. complete
43. exit
44. active-cac-policy-set
## Chapter 49      Late-to-Early Media Interworking

### Configuring Late-to-Early Media Interworking

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>configure terminal</code></td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>sbc service-name</code></td>
<td>Enters the submode for configuring the method profile.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Use the <code>service-name</code> argument to define the name of the</td>
</tr>
<tr>
<td>Router(config)# sbc mysbc</td>
<td>service.</td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>sbe</code></td>
<td>Enters the mode of an SBE entity within an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> <code>adjacency sip</code></td>
<td>Configures an adjacency.</td>
</tr>
<tr>
<td><code>adjacency-name</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe)# adjacency</td>
<td></td>
</tr>
<tr>
<td>sip sipGW</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> <code>nat force-off</code></td>
<td>Configures a SIP adjacency to assume that all endpoints are</td>
</tr>
<tr>
<td><code>preferred-transport udp</code></td>
<td>behind a NAT device.</td>
</tr>
<tr>
<td><code>redirect-mode pass-through</code></td>
<td>Sets the preferred transport protocol for SIP signaling on</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>an adjacency.</td>
</tr>
<tr>
<td>Router(config-sbc-sbe-adj-sip)#</td>
<td></td>
</tr>
<tr>
<td>preferred-transport udp</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> <code>authentication nonce</code></td>
<td>Configures the behavior of SBC on receipt of a 3xx</td>
</tr>
<tr>
<td><code>timeout value</code></td>
<td>response to an invite from the SIP adjacency.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-adj-sip)#</td>
<td></td>
</tr>
<tr>
<td>authentication nonce timeout 10</td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong> <code>signaling-address</code></td>
<td>Defines the local IPv4 signaling address of a SIP adjacency.</td>
</tr>
<tr>
<td><code>ipv4</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-adj-sip)#</td>
<td></td>
</tr>
<tr>
<td>signaling-address ipv4 10.10.10.10</td>
<td></td>
</tr>
</tbody>
</table>
### Chapter 49      Late-to-Early Media Interworking

#### Configuring Late-to-Early Media Interworking

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 10</td>
<td><code>signaling-port</code></td>
<td>Defines the local port of signaling address of a SIP adjacency.</td>
</tr>
<tr>
<td></td>
<td><code>signaling-port</code></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-adj-sip)# signaling-port 5000</td>
<td></td>
</tr>
<tr>
<td>Step 11</td>
<td><code>remote-address ipv4</code></td>
<td>Configures a SIP adjacency to restrict the set of remote signaling peers that can be contacted over the adjacency to those with the given IP address prefix.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-adj-sip)# remote-address ipv4 36.36.36.20 255.255.255.0</td>
<td></td>
</tr>
<tr>
<td>Step 12</td>
<td><code>signaling-peer</code></td>
<td>Configures a SIP adjacency to use the given remote signaling-peer.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-adj-sip)# signaling-peer gk andrew</td>
<td></td>
</tr>
<tr>
<td>Step 13</td>
<td><code>signaling-peer-port</code></td>
<td>Configures a SIP adjacency to use the given remote signaling-peer's port.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-adj-sip)# signaling-peer-port 123</td>
<td></td>
</tr>
<tr>
<td>Step 14</td>
<td><code>dbe-location-id</code></td>
<td>Configures an adjacency to use a given media gateway DBE location when routing media.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-adj-sip)# dbe-location-id 1</td>
<td></td>
</tr>
<tr>
<td>Step 15</td>
<td><code>account</code></td>
<td>Defines a SIP adjacency account on an SBE.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-adj-sip)# account isp42</td>
<td></td>
</tr>
<tr>
<td>Step 16</td>
<td><code>reg-min-expiry</code></td>
<td>Configures the minimum registration period in seconds on the SIP adjacency.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-adj-sip)# reg-min-expiry 300</td>
<td></td>
</tr>
<tr>
<td>Step 17</td>
<td>`media-late-to-early-iw {incoming</td>
<td>outgoing}`</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-adj-sip)# media-late-to-early-iw incoming</td>
<td></td>
</tr>
<tr>
<td>Step 18</td>
<td><code>attach</code></td>
<td>Attaches an adjacency to an account on an SBE.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-adj-sip)# attach</td>
<td></td>
</tr>
</tbody>
</table>
### Configuring Late-to-Early Media Interworking

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 19</strong> exit</td>
<td>Exits the current configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-adj-sip)#</td>
<td></td>
</tr>
<tr>
<td>exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 20</strong> exit</td>
<td>Exits the current configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-adj)#</td>
<td></td>
</tr>
<tr>
<td>exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 21</strong> sip inherit profile</td>
<td>Configures a global inherit profile.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe)#</td>
<td></td>
</tr>
<tr>
<td>sip inherit profile preset-p-cscf-access</td>
<td></td>
</tr>
<tr>
<td><strong>Step 22</strong> cac-policy-set</td>
<td>Enters the submode of CAC policy set configuration within</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>an SBE entity.</td>
</tr>
<tr>
<td>Router(config-sbc-sbe)#</td>
<td></td>
</tr>
<tr>
<td>cac-policy-set 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 23</strong> first-cac-table</td>
<td>Configures the name of the first policy table to process</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>when performing the admission control stage of policy.</td>
</tr>
<tr>
<td>Router(config-sbc-sbe-cacpolicy)#</td>
<td></td>
</tr>
<tr>
<td>first-cac-table RootCacTable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 24</strong> first-cac-scope</td>
<td>Configures the scope at which to begin defining limits</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>when performing the admission control stage of policy.</td>
</tr>
<tr>
<td>Router(config-sbc-sbe-cacpolicy)#</td>
<td></td>
</tr>
<tr>
<td>first-cac-scope src-adjacency</td>
<td></td>
</tr>
<tr>
<td><strong>Step 25</strong> averaging-period</td>
<td>Configures the size of the averaging period used by CAC</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>for its rate calculations.</td>
</tr>
<tr>
<td>Router(config-sbc-sbe-cacpolicy)#</td>
<td></td>
</tr>
<tr>
<td>averaging-period 5</td>
<td></td>
</tr>
<tr>
<td><strong>Step 26</strong> cac-table</td>
<td>Creates or configures an admission control table.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-cacpolicy)#</td>
<td></td>
</tr>
<tr>
<td>cac-table MyCacTable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 27</strong> table-type limit list of limit tables</td>
<td>Configures a CAC Limit table type.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe-cacpolicy-cactable)#</td>
<td></td>
</tr>
<tr>
<td>table-type limit call-priority</td>
<td></td>
</tr>
</tbody>
</table>
### Step 28: `entry num`

**Example:**
```
Router(config-sbc-sbe-cacpolicy-cactable)#
entry 1
```

**Purpose:** Creates or modifies an entry in a table.

### Step 29: `match-value value-keyword`

**Example:**
```
Router(config-sbc-sbe-cacpolicy-cactable-entry)#
match-value routine
```

**Purpose:** Configures the match-value of an entry in an admission control table. Use the ? to see a list of valid keywords.

### Step 30: `action cac-complete`

**Example:**
```
Router(config-sbc-sbe-cacpolicy-cactable-entry)#
action cac-complete
```

**Purpose:** Specifies that when an event matches, this CAC policy is complete.

### Step 31: `max-bandwidth`

**Example:**
```
Router(config-sbc-sbe-cacpolicy-cactable-entry)#
max-bandwidth 6000000
```

**Purpose:** Configures the maximum bandwidth for an entry in an admission control table.

### Step 32: `max-updates`

**Example:**
```
Router(config-sbc-sbe-cacpolicy-cactable-entry)#
max-updates 500
```

**Purpose:** Configures the maximum call updates for an entry in an admission control table.

### Step 33: `max-channels`

**Example:**
```
Router(config-sbc-sbe-cacpolicy-cactable-entry)#
max-channels 50
```

**Purpose:** Configures the maximum number of channels for an entry in an admission control table.

### Step 34: `early-media-type {backward-half-duplex | forward-half-duplex | full-duplex}`

**Example:**
```
Router(config-sbc-sbe-cacpolicy-cactable-entry)#
early-media-type full-duplex
```

**Purpose:** Configures the direction of early media to allow for an entry in a call admission control table.

### Step 35: `early-media-timeout`

**Example:**
```
Router(config-sbc-sbe-cacpolicy-cactable-entry)#
early-media-timeout 90
```

**Purpose:** Configures the amount of time for which to allow early-media before a call is established.

### Step 36: `codec-restrict-to-list`

**Example:**
```
Router(config-sbc-sbe-cacpolicy-cactable-entry)#
codec-restrict-to-list my_codecs
```

**Purpose:** Configures the CAC to restrict the codecs used in signaling a call to the set of codecs given in the named list.
### Command or Action

<table>
<thead>
<tr>
<th>Step 37</th>
<th>caller-codec-list</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-cacpolicy-cactable-entry) # caller-codec-list test</td>
</tr>
<tr>
<td>Purpose</td>
<td>Lists the codecs which the caller leg of a call is allowed to use.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 38</th>
<th>callee-privacy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-cacpolicy-cactable-entry) # callee-privacy always</td>
</tr>
<tr>
<td>Purpose</td>
<td>Configures the level of privacy processing to perform on messages sent from callee to caller.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 39</th>
<th>caller-privacy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-cacpolicy-cactable-entry) # caller-privacy always</td>
</tr>
<tr>
<td>Purpose</td>
<td>Configures the level of privacy processing to perform on messages sent from caller to callee.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 40</th>
<th>exit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-cacpolicy-cactable-entry) # exit</td>
</tr>
<tr>
<td>Purpose</td>
<td>Exits the current configuration mode.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 41</th>
<th>exit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-cacpolicy-cactable)# exit</td>
</tr>
<tr>
<td>Purpose</td>
<td>Exits the current configuration mode.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 42</th>
<th>complete</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-cacpolicy-cactable)# complete</td>
</tr>
<tr>
<td>Purpose</td>
<td>Completes the CAC-policy or call-policy set after committing the full set.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 43</th>
<th>exit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-cacpolicy)# exit</td>
</tr>
<tr>
<td>Purpose</td>
<td>Exits the current configuration mode.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 44</th>
<th>active-cac-policy-set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe-cacpolicy)# active-cac-policy-set 1</td>
</tr>
<tr>
<td>Purpose</td>
<td>Sets the active CAC-policy-set within an SBE entity.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 45</th>
<th>show sbc sbc-name sbe sip essential-methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Router(config-sbc-sbe)# show sbc mysbc sbe sip essential-methods</td>
</tr>
<tr>
<td>Purpose</td>
<td></td>
</tr>
</tbody>
</table>
Configuring Customized Offers for Late-to-Early Media Interworking

Prerequisites

Before performing this task, configure late-to-early media interworking per adjacency.

SUMMARY STEPS

1. **configure terminal**
2. **sbc service-name**
3. **sbe**
4. **sip sdp-media-profile** *profile-name*
5. **entry** *entry-num*
6. **media-line** *index* "*media_description*"
7. (Optional) Repeat the previous step with a different *index* to add more media lines to this entry.
8. **exit**
9. (Optional) Repeat Steps 6 through 9 with a different *entry-num* in Step 6 to add another entry to this profile.
10. **exit**
11. **exit**
12. **cac-policy-set** *policy-set-id*
13. **cac-table** *cac-table-name*
14. **entry** *entry-number*
15. **sip sdp-media-profile** *profile-name*
16. **Ctrl Z**
17. **show sbc** *sbc-name* **sbe** **sip sdp-media-profile** *profile-name*

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td></td>
</tr>
<tr>
<td><strong>configure terminal</strong></td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td></td>
</tr>
<tr>
<td><strong>sbc</strong> <em>service-name</em></td>
<td>Enters the submode for configuring the method profile. Use the <em>service-name</em> argument to define the name of the service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# sbc mysbc</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td></td>
</tr>
<tr>
<td><strong>sbe</strong></td>
<td>Enters the mode of an SBE entity within an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc)# sbe</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Step 4</th>
<th><code>sip sdp-media-profile profile-name</code></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td><code>Router(config-sbc-sbe)# sip sdp-media-profile profile1</code></td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>Configures an SDP media profile for a customized offer. Enter into SIP SDP media profile configuration mode.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 5</th>
<th><code>entry sequence-num</code></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td><code>Router(config-sbc-sbe-sip-sdp-media)# entry 1</code></td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>Enters the submode for adding a section of media description to the profile. A section, or entry, can contain one or more media description lines.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 6</th>
<th><code>media-line index &quot;media_description&quot;</code></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td><code>Router(config-sbc-sbe-sip-sdp-media-ele)# media-line 1 &quot;m=audio 0 RTP/AVP 0&quot;</code></td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>Adds a media description line to the entry. Quotation marks must surround the media description. See &quot;$paranum&gt;Rules for Media Lines in SDP Media Profiles? section on page 49-3.&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 7</th>
<th>(Optional) Repeat the previous step with a different <code>index</code> to add more media lines to this entry.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td><code>Router(config-sbc-sbe-sip-sdp-media-ele)# media-line 2 &quot;a=rtpmap:12 H264/90000&quot;</code></td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>Adds additional media descriptions to the entry. The index controls the ordering of the media descriptions.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 8</th>
<th><code>exit</code></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td><code>Router(config-sbc-sbe-sip-sdp-media-ele)# exit</code></td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>Exits the current configuration mode.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 9</th>
<th>(Optional) Repeat Steps 5 through 8 with a different <code>entry-num in Step 5.</code></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td><code>Router(config-sbc-sbe-sip-sdp-media-ele)# entry 2</code></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>Router(config-sbc-sbe-sip-sdp-media-ele)# media-line 1 &quot;m=audio 0 RTP/AVP 0&quot;</code></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>Router(config-sbc-sbe-sip-sdp-media-ele)# media-line 1 &quot;m=audio 0 RTP/AVP 0&quot;</code></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>Router(config-sbc-sbe-sip-sdp-media-ele)# media-line 2 &quot;a=rtpmap:0 PCMU/8000&quot;</code></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>Router(config-sbc-sbe-sip-sdp-media-ele)# exit</code></td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>Adds another entry to this profile.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 10</th>
<th><code>exit</code></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td><code>Router(config-sbc-sbe-sip-sdp-media)# exit</code></td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>Exits the current configuration mode.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 11</th>
<th><code>exit</code></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td><code>Router(config-sbc-sbe-sip)# exit</code></td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>Exits the current configuration mode.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 12</th>
<th><code>cac-policy-set policy-set-id</code></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td><code>Router(config-sbc-sbe)# cac-policy-set 1</code></td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>Enters the submode to make a change to a previously configured CAC policy set. Changes are not permitted to the active policy set.</td>
</tr>
</tbody>
</table>
Configuration Examples for the Late-to-Early Media Interworking Feature

This section includes the following examples:

- Example: Late-to-Early Media Interworking, page 49-13
- Example: Customized Offer for Late-to-Early Media Interworking, page 49-16

Example: Late-to-Early Media Interworking

The following example shows a configuration of the Late-to-Early Media Interworking feature.

Note: The caller and callee commands have been used in this procedure. In some scenarios, the branch command can be used as an alternative to the caller and callee command pair. The branch command has been introduced in Release 3.5.0. See the ?Sparanum>Configuring Directed Nonlimiting CAC Policies? section on page 7-37 for information about this command.
Router(config)# sbc mySbc
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency sip SIPP-1
Router(config-sbc-sbe-adj-sip)# nat force-off
Router(config-sbc-sbe-adj-sip)# preferred-transport udp
Router(config-sbc-sbe-adj-sip)# redirect-mode pass-through
Router(config-sbc-sbe-adj-sip)# authentication nonce timeout 300
Router(config-sbc-sbe-adj-sip)# signaling-address ipv4 201.201.201.20
Router(config-sbc-sbe-adj-sip)# signaling-port 5060
Router(config-sbc-sbe-adj-sip)# signaling-peer-port 5060
Router(config-sbc-sbe-adj-sip)# dbe-location-id 4294967295
Router(config-sbc-sbe-adj-sip)# account SIPP-1
Router(config-sbc-sbe-adj-sip)# reg-min-expiry 3000
Router(config-sbc-sbe-adj-sip)# media-late-to-early-iw incoming
Router(config-sbc-sbe-adj-sip)# attach
Router(config-sbc-sbe-adj-sip)# exit
Router(config-sbc-sbe-adj)# exit
Router(config-sbc-sbe)# adjacency sip SIPP-2
Router(config-sbc-sbe-adj-sip)# nat force-off
Router(config-sbc-sbe-adj-sip)# preferred-transport udp
Router(config-sbc-sbe-adj-sip)# redirect-mode pass-through
Router(config-sbc-sbe-adj-sip)# authentication nonce timeout 300
Router(config-sbc-sbe-adj-sip)# signaling-address ipv4 201.201.201.20
Router(config-sbc-sbe-adj-sip)# signaling-port 5060
Router(config-sbc-sbe-adj-sip)# signaling-peer-port 5060
Router(config-sbc-sbe-adj-sip)# dbe-location-id 4294967295
Router(config-sbc-sbe-adj-sip)# account SIPP-2
Router(config-sbc-sbe-adj-sip)# reg-min-expiry 3000
Router(config-sbc-sbe-adj-sip)# media-late-to-early-iw outgoing
Router(config-sbc-sbe-adj-sip)# attach
Router(config-sbc-sbe-adj-sip)# exit
Router(config-sbc-sbe-adj)# exit
Router(config-sbc-sbe)# sip inherit profile preset-core
Router(config-sbc-sbe)# cac-policy-set 1
Router(config-sbc-sbe-cacpolicy)# first-cac-table table
Router(config-sbc-sbe-cacpolicy)# first-cac-scope call
Router(config-sbc-sbe-cacpolicy)# averaging-period 60
Router(config-sbc-sbe-cacpolicy)# cac-table table
Router(config-sbc-sbe-cacpolicy-cactable)# table-type limit adjacency
Router(config-sbc-sbe-cacpolicy-cactable)# entry 1
Router(config-sbc-sbe-cacpolicy-cactable-actable-entry)# match-value SIPP-1
Router(config-sbc-sbe-cacpolicy-cactable-actable-entry)# action cac-complete
Router(config-sbc-sbe-cacpolicy-cactable-actable-entry)# max-bandwidth 64009 Gbps
Router(config-sbc-sbe-cacpolicy-cactable-actable-entry)# max-updates 4294967295
Router(config-sbc-sbe-cacpolicy-cactable-actable-entry)# max-channels 4294967295
Router(config-sbc-sbe-cacpolicy-cactable-actable-entry)# early-media-type full-duplex
Router(config-sbc-sbe-cacpolicy-cactable-actable-entry)# early-media-timeout 0
Router(config-sbc-sbe-cacpolicy-cactable-actable-entry)# codec-restrict-to-list allowed_caller
Router(config-sbc-sbe-cacpolicy-cactable-actable-entry)# caller-codec-list allowed_caller
Router(config-sbc-sbe-cacpolicy-cactable-actable-entry)# callee-privacy never
Router(config-sbc-sbe-cacpolicy-cactable-actable-entry)# caller-privacy never
Router(config-sbc-sbe-cacpolicy-cactable-actable-entry)# entry 2
Router(config-sbc-sbe-cacpolicy-cactable-actable-entry)# match-value SIPP-2
Router(config-sbc-sbe-cacpolicy-cactable-actable-entry)# action cac-complete
Router(config-sbc-sbe-cacpolicy-cactable-actable-entry)# max-bandwidth 64009 Gbps
Router(config-sbc-sbe-cacpolicy-cactable-actable-entry)# max-updates 4294967295
Router(config-sbc-sbe-cacpolicy-cactable-actable-entry)# max-channels 4294967295
Router(config-sbc-sbe-cacpolicy-cactable-actable-entry)# early-media-type full-duplex
Router(config-sbc-sbe-cacpolicy-cactable-actable-entry)# early-media-timeout 0
Router(config-sbc-sbe-cacpolicy-cactable-actable-entry)# codec-restrict-to-list allowed
```config
Router(config-sbc-sbe-cacpolicy-cactable-entry)# callee-codec-list allowed
Router(config-sbc-sbe-cacpolicy-cactable-entry)# callee-privacy never
Router(config-sbc-sbe-cacpolicy-cactable-entry)# caller-privacy never
Router(config-sbc-sbe-cacpolicy-cactable-entry)# exit
Router(config-sbc-sbe-cacpolicy)# complete
Router(config-sbc-sbe-cacpolicy)# exit
Router (config-sbc-sbe)# active-cac-policy-set 1
Router(config-sbc-sbe)# retry-limit 3
Router(config-sbc-sbe)# call-policy-set 1
Router(config-sbc-sbe-rtgpolicy)# first-call-routing-table start-table
Router(config-sbc-sbe-rtgpolicy)# rtg-src-adjacency-table start-table
Router(config-sbc-sbe-rtgpolicy-entry)# entry 1
Router(config-sbc-sbe-rtgpolicy-entry)# action complete
Router(config-sbc-sbe-rtgpolicy-entry)# dst-adjacency SIPP-1
Router(config-sbc-sbe-rtgpolicy-entry)# match-adjacency SIPP-2
Router(config-sbc-sbe-rtgpolicy-entry)# exit
Router(config-sbc-sbe-rtgpolicy)# entry 2
Router(config-sbc-sbe-rtgpolicy-entry)# action complete
Router(config-sbc-sbe-rtgpolicy-entry)# dst-adjacency SIPP-2
Router(config-sbc-sbe-rtgpolicy-entry)# match-adjacency SIPP-1
Router(config-sbc-sbe-rtgpolicy-entry)# exit
Router(config-sbc-sbe-rtgpolicy)# complete
Router(config-sbc-sbe-rtgpolicy)# exit
Router(config-sbc-sbe)# active-call-policy-set 1
Router(config-sbc-sbe)# sip max-connections 2
Router(config-sbc-sbe)# sip timer
Router(config-sbc-sbe-tmr)# tcp-idle-timeout 120000
Router(config-sbc-sbe-tmr)# tls-idle-timeout 3600000
Router(config-sbc-sbe-tmr)# udp-response-linger-period 32000
Router(config-sbc-sbe-tmr)# udp-first-retransmit-interval 500
Router(config-sbc-sbe-tmr)# udp-max-retransmit-interval 4000
Router(config-sbc-sbe-tmr)# invite-timeout 180
Router(config-sbc-sbe-tmr)# exit
Router(config-sbc-sbe)# codec-list allowed
Router(config-sbc-sbe-codec-list)# description allowed codecs
Router(config-sbc-sbe-codec-list)# codec PCMA
Router(config-sbc-sbe-codec-list)# codec PCMU
Router(config-sbc-sbe-codec-list)# exit
Router(config-sbc-sbe)# codec-list allowed_caller
Router(config-sbc-sbe-codec-list)# description caller
Router(config-sbc-sbe-codec-list)# codec PCMA
Router(config-sbc-sbe-codec-list)# exit
Router(config-sbc-sbe)# h323
Router(config-sbc-sbe-h323)# ras timer arq 5000
Router(config-sbc-sbe-h323)# ras timer brq 3000
Router(config-sbc-sbe-h323)# ras timer drq 3000
Router(config-sbc-sbe-h323)# ras timer grq 5000
Router(config-sbc-sbe-h323)# ras timer urq 3000
Router(config-sbc-sbe-h323)# ras timer ttl 60
Router(config-sbc-sbe-h323)# h225 timer proceeding 10000
Router(config-sbc-sbe-h323)# h225 timer establishment 180000
Router(config-sbc-sbe-h323)# h225 timer setup 4000
Router(config-sbc-sbe-h323)# exit
Router(config-sbc-sbe)# h323
Router(config-sbc-sbe-h323)# adjacency timeout 30000
Router(config-sbc-sbe-h323)# exit
```
Example: Customized Offer for Late-to-Early Media Interworking

The following example configures a customized media description and assigns it to a CAC policy.

Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# sip sdpm-media-profile MediaProfile
Router(config-sbc-sbe-sip-sdp-media)# entry 1
Router(config-sbc-sbe-sip-sdp-media-ele)# media-line 1 "m=audio 0 RTP/AVP 31"
Router(config-sbc-sbe-sip-sdp-media-ele)# media-line 2 "a=aaa:testing"
Router(config-sbc-sbe-sip-sdp-media-ele)# Ctrl Z
Router# show sbc test sbe sip sdpm-media-profile MediaProfile
SDP media profile "MediaProfile"
  Elements:
    Sequence Number : 1
    media-line 1 : m=audio 0 RTP/AVP 31
    media-line 2 : a=aaa:testing

Not in use by any CAC table entries

Router# configure terminal
Router(config)# sbc test
Router(config-sbc)# sbe
Router(config-sbc-sbe)# cac-policy-set 1
Router(config-sbc-sbe-cacpolicy)# cac-table testpolicytable
Router(config-sbc-sbe-cacpolicy-cactable)# entry 1
Router(config-sbc-sbe-cacpolicy-cactable-entry)# sip sdpm-media-profile MediaProfile
Router(config-sbc-sbe-cacpolicy-cactable-entry)# Ctrl Z
Router# show sbc test sbe sip sdpm-media-profile MediaProfile
SDP media profile "MediaProfile"
  Elements:
    Sequence Number : 1
    media-line 1 : m=audio 0 RTP/AVP 31
    media-line 2 : a=aaa:testing

In use by CAC table testpolicytable, entry 1
Verification

Use the commands listed in Table 49-1 to verify operation.

**Table 49-1 Commands to Verify Operation**

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>show sbc sbc-name sbe cac-policy-set id table name entries</code></td>
<td>Lists a summary of the CAC policy tables associated with the given policy set.</td>
</tr>
<tr>
<td><code>show sbc sbc-name sbe adjacencies</code></td>
<td>Lists the adjacencies configured on SBEs.</td>
</tr>
<tr>
<td><code>show sbc sbc-name sbe sdp-profiles</code></td>
<td>Lists the SIP SDP media profiles defined under a named SBE and indicates whether they are currently associated with a CAC policy.</td>
</tr>
<tr>
<td><code>show sbc sbc-name sbe sip sdp-media-profile [profile-name]</code></td>
<td>Lists the SIP SDP media profiles defined under a named SBE and indicates whether they are currently associated with a CAC policy, or, if you include a profile name, shows the contents of the named profile.</td>
</tr>
</tbody>
</table>

The following example shows adjacencies.

```
Router# show sbc test sbe adjacencies asrlk-1 de

SBC Service "test"
Adjacency asrlk-1 (SIP)
  Status: Attached
  Signaling address: 22.22.22.2:5060, VRF Admin
  Signaling-peer: 33.33.33.3:5060
  Remote address: 33.33.33.3 255.255.255.255
  Force next hop: No
  Account:
    Group: None
    In header profile: Default
    Out header profile: Default
    In method profile: Default
    Out method profile: Default
    In UA option prof: Default
    Out UA option prof: Default
    In proxy opt prof: Default
    Out proxy opt prof: Default
  Priority set name: None
  Local-id: None
  Rewrite REGISTER: Off
  Target address: None
  NAT Status: Auto Detect
  Reg-min-expiry: 3000 seconds
  Fast-register: Enabled
  Fast-register-int: 30 seconds
  Authenticated mode: None
  Authenticated realm: None
  Auth. nonce life time: 300 seconds
  IMS visited NetID: None
  Inherit profile: Default
  Force next hop: No
  Home network Id: None
  UnEncrypt key data: None
  SIPI passthrough: No
```

Router# show sbc test sbe adjacencies asr1k-2 de

SBC Service "test"
  Adjacency asr1k-2 (SIP)
    Status: Attached
    Signaling address: 22.22.22.2:5061, VRF Admin
    Signaling-peer: 44.44.44.4:5061
    Remote address: 44.44.44.4 255.255.255.255
    Force next hop: No
    Account: None
    Group: None
    In header profile: Default
    Out header profile: Default
    In method profile: Default
    Out method profile: Default
    In UA option prof: Default
    Out UA option prof: Default
    In proxy opt prof: Default
    Out proxy opt prof: Default
    Priority set name: None
    Local-id: None
    Rewrite REGISTER: Off
    Target address: None
    NAT Status: Auto Detect
    Reg-min-expiry: 3000 seconds
    Fast-register: Enabled
    Fast-register-int: 30 seconds
    Authenticated mode: None
    Authenticated realm: None
    Auth. nonce life time: 300 seconds
    IMS visited NetID: None
    Inherit profile: Default
    Force next hop: No
    Home network Id: None
    UnEncrypt key data: None
    SIPI passthrough: No
    Rewrite from domain: Yes
    Rewrite to header: Yes
    Media passthrough: No
    Hunting Triggers: Global Triggers
    Redirect mode: Pass-through
    Security: Untrusted
    Outbound-flood-rate: None
    Ping-enabled: No
    Signaling Peer Status: Not Tested
    media-late-to-early-lw: outgoing

The following command lists a summary of the CAC policy tables associated with the given policy set:

Router# show sbc test sbe cac-policy-set 1 table table entry 1

SBC Service "test"
Policy set 1 table table entry 1
Match value               SIPP-1
Action                    CAC policy complete
Max updates               Unlimited
Max bandwidth             Unlimited
Max channels              Unlimited
Transcoder                Allowed
Caller privacy setting    Never hide
Callee privacy setting    Never hide
Early media               Allowed
Early media direction     Both
Early media timeout       0
Caller voice QoS profile  default
Caller video QoS profile  default
Caller sig QoS profile    default
Callee voice QoS profile  default
Callee video QoS profile  default
Callee sig QoS profile    default
Restrict codecs to list   allowed_caller
Restrict caller codecs to list   default
Restrict callee codecs to list   allowed
Media bypass              Allowed
Number of calls rejected by this entry    0

Router# show sbc test sbe cac-policy-set 1 table table entry 2

SBC Service "test"
Policy set 1 table table entry 2
Match value               SIPP-2
Action                    CAC policy complete
Max updates               Unlimited
Max bandwidth             Unlimited
Max channels              Unlimited
Transcoder                Allowed
Caller privacy setting    Never hide
Callee privacy setting    Never hide
Early media               Allowed
Early media direction     Both
Early media timeout       0
Caller voice QoS profile  default
Caller video QoS profile  default
Caller sig QoS profile    default
Callee voice QoS profile  default
Callee video QoS profile  default
Callee sig QoS profile    default
Restrict codecs to list   allowed
Restrict caller codecs to list   default
Restrict callee codecs to list   allowed
Media bypass              Allowed
Number of calls rejected by this entry    0

Router#

The following example shows a list of SDP media profiles configured under an SBC service:

Router# show sbc test sbe sip sdmp-media-profile
SDP Media profiles for SBC service "test"

<table>
<thead>
<tr>
<th>Name</th>
<th>In use</th>
</tr>
</thead>
<tbody>
<tr>
<td>MediaProfile</td>
<td>Yes</td>
</tr>
</tbody>
</table>

The following example shows the contents of a named SDP media profile:

Router# show sbc test sbe sip sdmp-media-profile MediaProfile
SDP media profile 'MediaProfile'
   Elements:
   Sequence Number : 1
      media-Line 1      : m=audio 0 RTP/AVP 31
      media-Line 2      : a=aaa:testing

In use by CAC table testpolicytable, entry 1
Early Media

The Early Media feature is supported for Session Initiation Protocol (SIP) calls. Early Media is the ability of two user agents to communicate before a call is actually established. Support for early media is important both for interoperability with the Public Switched Telephone Network (PSTN) and billing purposes.

Early Media is defined when media begins to flow before the call is officially connected. Media channels are set up prior to the call connection. These channels are used to provide the ring tone that the caller hears and are not generated by the caller’s endpoint or other queuing services, for example, hold music.

Note

For Cisco IOS XR Software Release and later, this feature is supported in the unified model only.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).

Feature History for Early Media

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XR Software Release</td>
<td>This feature was introduced on the Cisco IOS XR along with support for the unified model.</td>
</tr>
</tbody>
</table>

Contents

This module contains the following sections:

- Restrictions for the Early Media Support, page 50-1
- Information About Early Media, page 50-2

Restrictions for the Early Media Support

The restrictions for Early Media Support are:

- Cisco Unified Border Element (SP Edition) offers support for the gateway model of early media (as defined in RFC 3960).
- Early media does not work with endpoints which send late SDP.
- Cisco Unified Border Element (SP Edition) does not currently support RFC 3312.
Information About Early Media

Current implementations support early media through the 183 response code. When the called party wishes to send early media to the caller, it sends a 183 response to the caller. This response contains the Session Description Protocol (SDP). When the caller receives the response, it suppresses any local alerting of the user (for example, audible ring tones or a pop-up window) and begins playing out the media that it receives. The SDP in the 183 response provides an address, to which the real-time control protocol (RTCP) packets can be sent.

Some implementations take media from the caller, and send it to the callee as well. If the call is ultimately rejected, the called party generates a non-2xx final response. When this response is received by the caller, it ceases playing out, or sending media. However, if the call is accepted, the called party generates a 2xx response (generally, with the same SDP as in the 183 response), and sends it to the caller. The media transmission continues as before.

In addition, Cisco Unified Border Element (SP Edition) supports the following for early media:

- Renegotiation of the media after early media is flowing (before and after the call is connected). Media renegotiation is supported on Cisco Unified Border Element (SP Edition) using the PRACK and UPDATE methods.
- Optional SIP UPDATE support by SIP endpoints (including early media without UPDATE support).
- RFC 3312 preconditions.
- Configurable SIP support of Required, Supported, and Proxy-Require headers.
- A per-adjacency flag to allow interoperability with the Cisco Gateway’s non-standard PRACK behavior.
SIP Instant Messaging

Cisco Unified Border Element (SP Edition) supports SIP instant messaging (IM). Two options for SIP instant messaging are configurable—record-route passthrough and privacy for SIP Instant Messaging. The typical SIP instant messaging implementation uses end-to-end Record-Route passthrough, but privacy is not applied.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).

For a complete description of the commands used in this chapter, refer to the Cisco Unified Border Element (SP Edition) Command Reference: Unified Model at:

For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

Feature History for SIP Instant Messaging

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 2.5</td>
<td>The SIP Instant Messaging feature was introduced on the Cisco IOS XR.</td>
</tr>
</tbody>
</table>

Contents

This chapter contains the following sections:

- SIP Instant Messaging, page 51-1
- Configurable Options for SIP Instant Messaging, page 51-2
- SDP Handling for SIP Instant Messaging, page 51-5

SIP Instant Messaging

Cisco Unified Border Element (SP Edition) supports SIP instant messaging (IM). For SIP instant messaging, SBC handles the calls as follows:

- Calls with the Media Announcement m=message or m=x-ms-message are allowed.
- Messages containing multiple m=message lines are permitted.
SBC does not allocate any bandwidth to support IM dialogs because there is no media stream for such dialogs.

- The IM message is rejected if the SDP also contains any other media types.
- SBC forwards the SDP body unchanged unless privacy is configured.

SBC can be configured to pass through Record-Route headers from one side of the IM dialog to the other so that the end-to-end Route header set is available to caller and callee. In this case, SBC remains in the flow of messages by appending Record-Route headers and not rewriting the Contact.

### Configurable Options for SIP Instant Messaging

Two options for SIP instant messaging are configurable and explained in the following sections:

- **Record-Route Passthrough for SIP Instant Messaging**, page 51-2
- **Privacy for SIP Instant Messaging**, page 51-4

The typical SIP instant messaging implementation will use end-to-end Record-Route passthrough but will not apply privacy.

### Record-Route Passthrough for SIP Instant Messaging

This section explains Record-Route passthrough and contains the following subsections:

- **Record-Route Passthrough Overview**, page 51-2
- **Registered Subscribers and Record-Route Passthrough**, page 51-3
- **Record-Route Passthrough Configuration**, page 51-3

#### Record-Route Passthrough Overview

A Record-Route set represents the hop-by-hop route SIP messages must traverse between two endpoints as part of a SIP dialog. The final hop to the endpoint is represented by the Contact header. SBC’s normal behavior is to rewrite the Contact and maintain two independent Record-Route sets, one for each side of the call. Each of these Record-Route sets represents the route between the endpoint and the adjacency on that side of the call.

Passing through the end-to-end Record-Route set means that the adjacencies on a call do not represent the last hop and therefore must not be identified by the Contact header. When passing through the end-to-end Record Route set, SBC also passes through the end-to-end Contacts without rewriting them. In this case, SBC remains in the flow by appending Record-Route headers representing the inbound and outbound adjacencies.

If inbound and outbound adjacencies have conflicting Record-Route passthrough configurations, the setting of the inbound adjacency is used. For example, if the inbound adjacency enables Record-Route passthrough but the outbound adjacency does not, the outbound adjacency will forward the supplied Record-Route set, append a Record-Route header corresponding to the outbound adjacency, and leave the Contact header unaltered.
Registered Subscribers and Record-Route Passthrough

Record-Route passthrough behavior does not apply to SIP messages received from or going to Registered endpoints regardless of the inbound adjacency’s Record-Route passthrough configuration. The SIP specification (RFC 326) mandates that registrars ignore Record-Route headers present on a REGISTER message. Therefore to ensure that SBC remains in subsequent dialogs created to or from registered subscribers, SBC must rewrite the Contact in REGISTER messages. SBC updates the Contacts in later dialogs created by a registered endpoint so that they match the Contact previously published to network. At the point that SBC rewrites the Contact, SBC terminates any existing Record-Route set and creates a new one.

Record-Route Passthrough Configuration

To pass through Record-Route sets, SBC provides configuration on a per adjacency basis using the passthrough header record-route command.

- If turned off (no passthrough header record-route), the Record-Route set is cached and returned to the endpoint. Thereafter, the Record-Route set is used to build Route headers on subsequent outgoing messages. Neither end will see the entire end-to-end Record-Route set. This is the default behavior.

- If turned on, the request is not from or to a registered subscriber, the following occurs:
  - For dialog-creating requests, any Record-Route set present on the request is passed through SBC and forwarded to the receiving endpoint.
  - The Contact present on the request is passed through SBC and forwarded to the receiving endpoint.
  - Record-Route headers representing the inbound and outbound adjacency are appended to the request.
  - The end-to-end Record-Route set is passed back through SBC on the response and forwarded to the calling endpoint. Both endpoints see the entire end-to-end Record-Route set.
  - The preceding will be the behavior regardless of outbound adjacency configuration.
  - Subsequent in-dialog requests do not have their Request URI updated to match the Contact received on the dialog-creating request.

If a request is from or to a registered subscriber, it is processed as though record-route passthrough was turned off.

If topology hiding or privacy is applied to a call, the Record-Route set is stripped from the request regardless of the record-route passthrough configuration.

Configuring Record-Route passthrough does not result in the Route headers being passed through on subsequent messages.

With Record-Route passthrough enabled, messages in an IM dialog are adjusted as described earlier in this section. As an example of these Record-Route passthrough adjustments, the following requests show how enabling Record-Route passthrough affects an INVITE request that is used for an IM dialog. In the first example, Record-Route passthrough is not used. In the second example, Record-Route passthrough is enabled, and SBC adds the Record-Route information in bold font to the INVITE request. In both examples, privacy is not used.

In these examples, the INVITE requests are outbound from the SBC to a SIP proxy server.

Outbound INVITE (without Record-Route passthrough):

```
INVITE sip:callee@callee.com SIP/2.0
```

Outbound INVITE (with Record-Route passthrough):

```
INVITE sip:callee@callee.com SIP/2.0
```
Privacy for SIP Instant Messaging

For IM privacy, the CAC configuration that is used for normal calls is also used for IM dialogs. The CAC table entry command option `caller-privacy` configures privacy. For information on using this command, see the Cisco Unified Border Element (SP Edition) Command Reference: Unified Model at http://www.cisco.com/en/US/docs/ios/sbc/command/reference/sbcu_book.html.

If privacy is enabled, SBC adjusts three fields in the SIP IM SDP to make them anonymous.

- The Connection lines (c=) may contain sensitive IP address information in the connection address field. This information is hidden by inserting a Session Description level Connection line with the local address of the corresponding adjacency. Any other Connection lines are removed.

- The o= line is replaced with SBC's own information. The address field in this line is set to match the c= line.

- For SDP indicating a SIP IM stream, the Media Descriptions lines (m=) will be of the form:

  \[ m\text{-}message\ \text{port[\text{number_of_ports}] \text{sip} \text{format_list} } \]

  In the preceding, `format_list` is a SIP URL and may contain user sensitive information. The user-sensitive information is hidden by stripping all messages indicating Media Description lines and forwarding an Offer containing the following Media Description line:

  \[ m\text{-}message\ 5060\ \text{sip}\ \text{sip:anonymous@192.168.10.10} \]

The following examples show how enabling `caller-privacy` affects the SDP messages that are used for an IM dialog. In the examples, 192.168.10.10 is the local address of the outbound adjacency, and 192.168.2.20 is the address of the inbound adjacency. In the examples, the information in bold font has been adjusted to make the caller anonymous. The following example is outbound from the SBC to a SIP proxy server.

Outbound INVITE (without privacy):

\[ v=0 \]

Outbound INVITE (with Record-Route passthrough):

\[ v=0 \]

Outbound INVITE (with Record-Route passthrough):

\[ v=0 \]
SDP Handling for SIP Instant Messaging

In the SDP, SBC identifies IM dialogs by the presence of m=message or m=x-ms-message. For an offer that includes both IM and audio/video media lines, SBC rejects the offer with error code 488, Not Acceptable Here. After the initial media negotiation, any subsequent reoffer that attempts to change a call from an IM to a non-IM or vice versa is also rejected with error code 488.

SBC passes through SDP for IM dialogs unchanged with the exception of modifications due to privacy (see Privacy for SIP Instant Messaging? section on page 51-4). SBC does not allocate any gates or media pinholes.

The following sections provide more information on how SBC handles SDP for SIP instant messaging:

- SIP URLs in the SDP? section on page 51-6
- Miscellaneous SDP Handling? section on page 51-6
SIP URLs in the SDP

SDP attached to SIP IM dialogs can contain SIP URLs in the Media Description lines. For example:

```
m=message 5060 sip sip:example@home.net
```

If an endpoint sends an in-dialog message to the URL in the SDP (by placing the URL in the Request URI), the behavior is as follows:

- If SBC has not been configured to pass through the Record-Route set, the request may fail to route.
- If SBC has been configured to pass through the Record-Route set, the request will be forwarded by SBC without altering the Request URI.

For information on Record-Route passthrough, see the "Record-Route Passthrough for SIP Instant Messaging? section on page 51-2.

Miscellaneous SDP Handling

This section describes some miscellaneous SDP handling for SIP instant messaging.

- Transcoding—Since there is no media stream, SBC never attempts to bring in a transcoder. If the callee endpoint does not support the media type, the call fails.
- Special Media Descriptions—In some cases, SDP may include a proprietary media type of m=x-ms-message. SBC treats m=x-ms-message exactly the same as m=message. No support is added for any other proprietary media types.
- Interworking—SIP instant messaging does not interwork for SIP/H.323 calls, or for SIP-SIP late to early media interworking calls. If an IM dialogs is used in these scenarios, call setup fails with the response code 488, Not Acceptable Here.
- Invalid Connection Address—SBC does not edit the SDP of SIP IM dialogs except when privacy is configured. Therefore, the connection address passed through SBC may not be valid as far as the receiving endpoint is concerned. This is acceptable because no media is flowing between the endpoints.
Integration of Resource Management and SIP

As per IETF RFC 3312, call endpoints can determine whether resources are fully reserved for a media stream before using it. This feature is useful when separate quality of service (QoS) signaling, such as Resource ReSerVation Protocol (RSVP), is used. To accomplish this, RFC 3312 defines three new a=lines at media stream granularity. Endpoints use these lines to signal reservation information and their preconditions for adopting the new Session Description Protocol (SDP).

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).

Note

For Cisco IOS XE Release 2.4 and later, this feature is supported in the unified model only.

Feature History for Integration of Resource Management and SIP Support

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 2.4</td>
<td>This feature was introduced on the Cisco IOS XR along with support for the unified model.</td>
</tr>
</tbody>
</table>

Contents

This module contains the following sections:

- Restrictions for Integration of Resource Management, page 52-1
- Information about Integration of Resource Management, page 52-2

Restrictions for Integration of Resource Management

The restrictions for integration of resource management are:

- When this feature is implemented, Cisco Unified Border Element (SP Edition) does not report the media state or generate preconditions. It only detects whether preconditions are present, and whether all the mandatory preconditions have been met if preconditions exist.
- This feature is a SIP-only feature and is not supported by H.323 or SIP-H.323 interworking.
- With RFC 3312 signaling procedures, media renegotiation is completed only when the mandatory preconditions have been met.
Information about Integration of Resource Management

When the precondition tag appears in the Require or Supported header fields of SIP messages, Cisco Unified Border Element (SP Edition) allows them to pass through. Cisco Unified Border Element (SP Edition) also allows the unmodified SDP to pass through, which represents the state and the preconditions.

When processing an offer results in failure, the underlying SIP message is either rejected or the call is torn down. When processing an answer results in failure, the call is torn down, regardless of the reason for the failure.
ENUM Client

Cisco Unified Border Element (SP Edition) supports E.164 Number Mapping (ENUM).

Feature History for Implementing SNMP

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 3.1S</td>
<td>ENUM Client Feature was introduced.</td>
</tr>
</tbody>
</table>

Contents

- Information about ENUM Client Configuration, page 53-1
- Configuring ENUM Client, page 53-3
- Configuration Examples of ENUM Client Configuration, page 53-11

Information about ENUM Client Configuration

E.164 Number Mapping (ENUM) is an IETF standard protocol for converting telephone numbers into IP addresses (and vice versa), so that the telephone numbers can be maintained by a DNS server.

The SBC ENUM client is configurable and accepts the ITU standard format for international telephone numbers, E.164: country code, area code, phone number.

The ENUM client translates telephone numbers into standard sip/sips URIs that are resolved by a DNS server and then stored in an SBC routing table. Currently, only IPv4 is supported.

When a telephone number is called, the ENUM client queries the DNS server for a sip/sips URI. The DNS server returns the URI to the ENUM client, and the ENUM client stores the URI in an SBC routing table.

Destination Address

The destination address of a called number is typically derived from the Request URI. However, the destination address may also be derived from other headers in the routing table, such as the To: header or the P-Called-Party-ID: header.
The ENUM Client feature provides the user with the ability to configure a prioritized list of headers. This list may consist of any non-essential SIP headers, including the To: header and the Request URI. Once the list is configured, SBC can derive destination addresses for called numbers from this list of headers.

Destination address headers are stored in the header filter profile MIB table. Destination addresses must conform to the address syntax specification defined in RFC 3261. An address header list may contain a maximum to 10 entries.

The ENUM Client first searches the Request URI. If it does not find a match for the called number, it then searches the header list.

**Source Address**

The source address of a calling party number is typically derived from the From: header. The source addresses can be modified using the following configuration.

```
header-profile <name>
  src-address
```

You can also configure a prioritized list of headers from which the source address for a calling number is derived. This list may consist of any non-essential SIP headers.

Source address headers are stored in the header filter profile MIB table. Source addresses must conform to the address syntax specification defined in RFC 3261. An address header list may contain a maximum to 10 entries.

**Diverted-by Address**

The ENUM Client feature also provides support for deriving the source number from a prioritized list of headers for calls which have been diverted by another number. If a call has been diverted by another number, the source address must be derived from the diverted-by list of headers. Users can also configure a header action to reject these types of calls.

**Header Profiles**

The user can configure actions to be performed on a target address by configuring a header profile.

The following actions can be configured in a header profile for a target address:

- goto-table-name
- complete
- reject

For the SBC ENUM client configuration steps, see the "Configuring ENUM Client" section on page 53-3.

For an example of SBC ENUM client configuration see the "Configuration Examples of ENUM Client Configuration" section on page 53-11.

Additionally, you can also configure the SIP DNS cache, using the following commands:

- cache lifetime—configures the lifetime of a cached DNS entry.
- cache limit—configures the maximum number of entries that are permitted in the cache.
Configuring ENUM Client

The sections presents two configurations:

- Configuring ENUM Client, page 53-3
- Configuring a Call Policy for Multiple ENUM Entries, page 53-4

Configuring ENUM Client

Use the following procedure to configure and ENUM client:

**SUMMARY STEPS**

1. configure terminal
2. sbc sbc-name
3. sbe
4. enum enum-id
5. req-timeout timeout
6. max-recursive-depth number
7. entry entry-name
8. server ipv4 ip_address [vrf vrf_name]
9. dial-plan-suffix suffix
10. max-responses number
11. activate
12. end

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
</tbody>
</table>

**Example:**

Router# configure terminal

<table>
<thead>
<tr>
<th><strong>Step 2</strong> sbc sbc-name</th>
<th>Creates the SBC service on Cisco Unified Border Element (SP Edition) and enters into SBC configuration mode.</th>
</tr>
</thead>
</table>

**Example:**

Router(config)# sbc MySBC

<table>
<thead>
<tr>
<th><strong>Step 3</strong> sbe</th>
<th>Enters the mode of the signaling border element (SBE) function of the SBC.</th>
</tr>
</thead>
</table>

**Example:**

Router(config-sbc)# sbe
## Configuring ENUM Client

### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td><code>enum enum-id</code></td>
<td>Assigns the ENUM CUSTOMER ID number and enters ENUM configuration mode. Currently, only the number 1 is allowed.</td>
</tr>
<tr>
<td>5</td>
<td><code>req-timeout timeout</code></td>
<td>Configures the ENUM request timeout period.</td>
</tr>
<tr>
<td>6</td>
<td><code>max-recursive-depth number</code></td>
<td>Configures the maximum number of recursive ENUM look-ups for non-terminal Resource Records (RR).</td>
</tr>
<tr>
<td>7</td>
<td><code>entry entry-name</code></td>
<td>Configures the ENUM Client entry name and enter the ENUM entry configuration mode.</td>
</tr>
<tr>
<td>8</td>
<td><code>server ipv4 ip_address [vrf vrf_name]</code></td>
<td>Configures the IPv4 address of a DNS server for ENUM Client and optionally associates the DNS server to a VRF.</td>
</tr>
<tr>
<td>9</td>
<td><code>dial-plan-suffix suffix</code></td>
<td>Configures the dial plan suffix used for the ENUM query.</td>
</tr>
<tr>
<td>10</td>
<td><code>max-responses number</code></td>
<td>Configures the maximum number of ENUM records returned to the routing module.</td>
</tr>
<tr>
<td>11</td>
<td><code>activate</code></td>
<td>Activates ENUM Client.</td>
</tr>
<tr>
<td>12</td>
<td><code>end</code></td>
<td>Exits configuration mode and returns to privileged EXEC mode.</td>
</tr>
</tbody>
</table>

### Configuring a Call Policy for Multiple ENUM Entries

Use the following procedure to configure a call policy for multiple ENUM entries:

1. Assigns the ENUM CUSTOMER ID number and enters ENUM configuration mode. Currently, only the number 1 is allowed.
2. Configures the ENUM request timeout period.
3. Configures the maximum number of recursive ENUM look-ups for non-terminal Resource Records (RR).
4. Configures the ENUM Client entry name and enter the ENUM entry configuration mode.
5. Configures the IPv4 address of a DNS server for ENUM Client and optionally associates the DNS server to a VRF.
6. Configures the dial plan suffix used for the ENUM query.
7. Configures the maximum number of ENUM records returned to the routing module.
8. Activates ENUM Client.
9. Exits configuration mode and returns to privileged EXEC mode.
SUMMARY STEPS

1. configure terminal
2. sbc sbc-name
3. sbe
4. enum enum-id
5. entry (enum) entry-name
6. server ipv4 ip_address [vrf vrf_name]
7. dial-plan-suffix suffix
8. entry (enum) entry-name
9. server ipv4 ip_address [vrf vrf_name]
10. dial-plan-suffix suffix
11. activate
12. exit
13. sip header-profile profile-name
14. dst-address
   or
   src-address
   or
   div-address
15. header-prio priority-level header-name header-name
16. exit
17. call-policy-set policy-set-id
18. first-call-routing-table table-name
19. rtg-src-adjacency-table table-id
20. entry entry-id
21. enum enum-id entry (enum) entry-name
22. action next-table goto-table-name
23. entry entry-id
24. match-adjacency key
25. enum enum-id entry (enum) entry-name
26. dst-adjacency target-adjacency
27. action complete
28. rtg-dst-address-table table-id
29. entry entry-id
30. match-address key
31. dst-adjacency target-adjacency
32. action complete
33. entry entry-id
### Configuring ENUM Client

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>34.</td>
<td>match-address key</td>
<td></td>
</tr>
<tr>
<td>35.</td>
<td>dst-adjacency target-adjacency</td>
<td></td>
</tr>
<tr>
<td>36.</td>
<td>action complete</td>
<td></td>
</tr>
<tr>
<td>37.</td>
<td>entry entry-id</td>
<td></td>
</tr>
<tr>
<td>38.</td>
<td>match-address key</td>
<td></td>
</tr>
<tr>
<td>39.</td>
<td>prefix</td>
<td></td>
</tr>
<tr>
<td>40.</td>
<td>dst-adjacency target-adjacency</td>
<td></td>
</tr>
<tr>
<td>41.</td>
<td>action complete</td>
<td></td>
</tr>
<tr>
<td>42.</td>
<td>complete</td>
<td></td>
</tr>
<tr>
<td>43.</td>
<td>end</td>
<td></td>
</tr>
</tbody>
</table>

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure terminal</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router# configure terminal</td>
</tr>
<tr>
<td>Enters global configuration mode.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>sbc sbc-name</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config)# sbc MySBC</td>
</tr>
<tr>
<td>Creates the SBC service on Cisco Unified Border Element (SP Edition) and enters into SBC configuration mode.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>sbe</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc)# sbe</td>
</tr>
<tr>
<td>Enters the mode of the signaling border element (SBE) function of the SBC.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>enum enum-id</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe)# enum 1</td>
</tr>
<tr>
<td>Assigns the ENUM ID number and enters ENUM configuration mode. Currently, only the number 1 is allowed.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>entry (enum) entry-name</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-enum)# entry default-enum</td>
</tr>
<tr>
<td>Configures the default ENUM entry and enters ENUM entry configuration mode.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>server ipv4 ip_address [vrf vrf_name]</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-enum-entry)# server ipv4 10.10.10.10</td>
</tr>
<tr>
<td>Configures the IPv4 address of a DNS server for the ENUM Client.</td>
<td></td>
</tr>
</tbody>
</table>
### Configuring ENUM Client

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td><code>dial-plan-suffix suffix</code></td>
<td>Configures the dial plan suffix used for this ENUM query.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-enum-entry)# dial-plan-suffix e164.arpa</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td><code>entry (enum) entry-name</code></td>
<td>Configures another ENUM entry and enters ENUM entry configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-enum-entry)# entry cisco-enum</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td><code>server ipv4 ip_address [vrf vrf_name]</code></td>
<td>Configures the IPv4 address of a DNS server for ENUM Client and associates the DNS server to a VRF.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-enum-entry)# server ipv4 10.0.0.22 vrf cisco-vrf</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td><code>dial-plan-suffix suffix</code></td>
<td>Configures the dial plan suffix used for this ENUM query.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-enum-entry)# dial-plan-suffix cisco.com</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td><code>activate</code></td>
<td>Activates the ENUM client.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-enum-entry)# activate</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td><code>exit</code></td>
<td>Exits to the previous mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-enum)# exit</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td><code>sip header-profile profile-name</code></td>
<td>Configures a header profile in the mode of an SBE entity.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe)# sip header-profile enum</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td><code>dst-address</code> or <code>src-address</code> or <code>div-address</code></td>
<td>Enters destination address submode. or Enters source address submode. or Enters diverted-by address submode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-sip-hdr)# dst-address or Router(config-sbc-sbe-sip-hdr)# src-address or Router(config-sbc-sbe-sip-hdr)# div-address</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
<td></td>
</tr>
</tbody>
</table>
| **Step 15** | **header-prio priority-level header-name**
  **header-name** |
  **Example:**
  Router(config-sbc-sbe-sip-hdr-dst)# header-prio 1 header-name Dst_Add_Hdr_1  
or  
  Router(config-sbc-sbe-sip-hdr-src)# header-prio 1 header-name Src_Add_Hdr_1  
or  
  Router(config-sbc-sbe-sip-hdr-div)# header-prio 1 header-name Div_Add_Hdr_1 |
  **Configures the priority of the header from which the destination address is derived.**
  **or**
  **Configures the priority of the header from which the source address is derived.**
  **or**
  **Configures the priority of the header from which the diverted-by address is derived.** |
| **Step 16** | **exit** |
  **Example:**
  Router(config-sbc-sbe-sip-hdr-src)# exit |
  **Exits to the previous mode.** |
| **Step 17** | **call-policy-set policy-set-id** |
  **Example:**
  Router(config-sbc-sbe-sip-hdr)# call-policy-set 1 |
  **Creates a new call policy set and enters SBE routing policy configuration mode.** |
| **Step 18** | **first-call-routing-table table-name** |
  **Example:**
  Router(config-sbc-sbe-rtgpolicy)# first-call-routing-table rt1 |
  **Configures the name of the first policy table to process when performing the routing stage of policy for new-call events.** |
| **Step 19** | **rtg-src-adjacency-table table-id** |
  **Example:**
  Router(config-sbc-sbe-rtgpolicy)# rtg-src-adjacency-table rt1 |
  **Enters the configuration mode of the existing routing table, in this case, rt1.** |
| **Step 20** | **entry entry-id** |
  **Example:**
  Router(config-sbc-sbe-rtgpolicy-rtgtable)# entry 2 |
  **Creates an entry in the routing table.** |
| **Step 21** | **enum enum-id entry (enum) entry-name** |
  **Example:**
  Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# enum 1 entry default-enum |
  **Configures the default ENUM entry for the routing table entry.** |
| **Step 22** | **action next-table goto-table-name** |
  **Example:**
  Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# action next-table dal |
  **Configures the action to take on routing table entry 1.** |
<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td>entry entry-id</td>
<td>Creates an entry in the routing table.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-rtgpolicy-rtgtable-entry) # entry 2</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>match-adjacency key</td>
<td>Configures the match value for entry 1 against a source adjacency. In this case, the source adjacency is sip2.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-rtgpolicy-rtgtable-entry) # match-adjacency sip2</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>enum enum-id entry (enum) entry-name</td>
<td>Configures an ENUM entry for the routing table entry.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-rtgpolicy-rtgtable-entry) # enum 1 entry cisco-enum</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>dst-adjacency target-adjacency</td>
<td>Configures the destination adjacency for entry 2 in table routing table.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-rtgpolicy-rtgtable-entry) # dst-adjacency sip-proxy1</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>action complete</td>
<td>Configures the action to take on routing table entry 2. In this case, the action is complete.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-rtgpolicy-rtgtable-entry) # action complete</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>rtg-dst-address-table table-id</td>
<td>Specifies the routing table (da1) that is searched for destination addresses to match called numbers.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-rtgpolicy-rtgtable-entry) # rtg-dst-address-table da1</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>entry entry-id</td>
<td>Creates an entry in the routing table.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-rtgpolicy-rtgtable) # entry 1</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>match-address key</td>
<td>Configures the match value for entry 1 in the routing table, to match against a destination called number.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-rtgpolicy-rtgtable-entry) # match-address bob</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>dst-adjacency target-adjacency</td>
<td>Configures the destination adjacency for entry 1 in table routing table.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config-sbc-sbe-rtgpolicy-rtgtable-entry) # dst-adjacency sip-proxy2</td>
<td></td>
</tr>
</tbody>
</table>
## Configuring ENUM Client

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 32</strong> action complete</td>
<td>Configures the action to take on routing table entry 1. In this case, the action is complete.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-rtgpolicy-rtgtable-entry) # action complete</td>
<td></td>
</tr>
<tr>
<td><strong>Step 33</strong> entry entry-id</td>
<td>Creates an entry in the routing table.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-rtgpolicy-rtgtable-entry) # entry 2</td>
<td></td>
</tr>
<tr>
<td><strong>Step 34</strong> match-address key</td>
<td>Configures the match value for entry 2 in the routing table, to match against a destination called number.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-rtgpolicy-rtgtable-entry) # match-address kate</td>
<td></td>
</tr>
<tr>
<td><strong>Step 35</strong> dst-adjacency target-adjacency</td>
<td>Configures the destination adjacency for entry 2 in table routing table.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-rtgpolicy-rtgtable-entry) # dst-adjacency sip-proxy3</td>
<td></td>
</tr>
<tr>
<td><strong>Step 36</strong> action complete</td>
<td>Configures the action to take on routing table entry 2. In this case, the action is complete.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-rtgpolicy-rtgtable-entry) # action complete</td>
<td></td>
</tr>
<tr>
<td><strong>Step 37</strong> entry entry-id</td>
<td>Creates an entry in the routing table.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-rtgpolicy-rtgtable-entry) # entry 3</td>
<td></td>
</tr>
<tr>
<td><strong>Step 38</strong> match-address key</td>
<td>Configures the match value for entry 3 in the routing table, to match against a destination called number.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-rtgpolicy-rtgtable-entry) # match-address 44</td>
<td></td>
</tr>
<tr>
<td><strong>Step 39</strong> prefix</td>
<td>Configures whether the match-address of this entry matches the start of the address.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-rtgpolicy-rtgtable-entry) # prefix</td>
<td></td>
</tr>
<tr>
<td><strong>Step 40</strong> dst-adjacency target-adjacency</td>
<td>Configures the destination adjacency for entry 3 in table routing table.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-rtgpolicy-rtgtable-entry) # dst-adjacency sip-proxy4</td>
<td></td>
</tr>
</tbody>
</table>
Example 1: ENUM Client

Use the following procedure to configure an ENUM Client:

```
Router# configure terminal
Router(config)# sbc MySBC
Router(config-sbc)# sbe
Router(config-sbc-sbe)# enum 1
Router(config-sbc-sbe-sbe)# req-timeout 10000
Router(config-sbc-sbe-sbe)# max-recursive-depth 100
Router(config-sbc-sbe-sbe)# entry ENUM_1
Router(config-sbc-sbe-sbe-sbe)# server ipv4 10.10.10.10 vrf VRF1
Router(config-sbc-sbe-sbe-sbe)# dial-plan-suffix Example.Suffix
Router(config-sbc-sbe-sbe-sbe)# max-responses 100
Router(config-sbc-sbe-sbe-sbe)# activate
Router(config-sbc-sbe-sbe-sbe)# end
```

Example:  
```
Router(config-sbc-sbe-sbe-sbe-sbe)# action complete
```

Step 41  
**action complete**  
Configures the action to take on routing table entry 3. In this case, the action is complete.

Example:  
```
Router(config-sbc-sbe-sbe-sbe-sbe)# action complete
```

Step 42  
**complete**  
Completes the call-policy set after committing the full set.

Example:  
```
Router(config-sbc-sbe-sbe-sbe-sbe)# complete
```

Step 43  
**end**  
Exits configuration mode and returns to privileged EXEC mode.

Example:  
```
Router(config-sbc-sbe-sbe-sbe-sbe)# end
```

Example 2: Call Policy for Multiple ENUM Entries

Use the following procedure to configure a call policy for multiple ENUM entries:

```
Router# configure terminal
Router(config)# sbc mysbc
Router(config-sbc)# sbe
Router(config-sbc-sbe)# enum 1
Router(config-sbc-sbe-sbe)# entry defaultEnum
Router(config-sbc-sbe-sbe-sbe)# server ipv4 192.168.10.1
Router(config-sbc-sbe-sbe-sbe-sbe)# dial-plan-suffix e164.arpa
Router(config-sbc-sbe-sbe-sbe-sbe)# entry ciscoEnum
Router(config-sbc-sbe-sbe-sbe-sbe)# server ipv4 10.0.0.22 vrf cisco-vrf
Router(config-sbc-sbe-sbe-sbe-sbe-sbe)# dial-plan-suffix cisco.com
Router(config-sbc-sbe-sbe-sbe-sbe-sbe)# activate
Router(config-sbc-sbe-sbe-sbe-sbe-sbe)# exit
```

Example:  
```
Router(config-sbc-sbe-sbe-sbe-sbe-sbe)# action complete
```

Step 41  
Configures the action to take on routing table entry 3. In this case, the action is complete.

Example:  
```
Router(config-sbc-sbe-sbe-sbe-sbe-sbe)# action complete
```

Step 42  
**complete**  
Completes the call-policy set after committing the full set.

Example:  
```
Router(config-sbc-sbe-sbe-sbe-sbe-sbe)# complete
```

Step 43  
**end**  
Exits configuration mode and returns to privileged EXEC mode.

Example:  
```
Router(config-sbc-sbe-sbe-sbe-sbe-sbe)# end
```

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 41 action complete</td>
<td>Configures the action to take on routing table entry 3. In this case, the action is complete.</td>
</tr>
<tr>
<td>Example: Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# action complete</td>
<td></td>
</tr>
<tr>
<td>Step 42 complete</td>
<td>Completes the call-policy set after committing the full set.</td>
</tr>
<tr>
<td>Example: Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# complete</td>
<td></td>
</tr>
<tr>
<td>Step 43 end</td>
<td>Exits configuration mode and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td>Example: Router(config-sbc-sbe-rtgpolicy)# end</td>
<td></td>
</tr>
</tbody>
</table>

Configuration Examples of ENUM Client Configuration

Example 1: ENUM Client

Use the following procedure to configure an ENUM Client:

```
Router# configure terminal
Router(config)# sbc MySBC
Router(config-sbc)# sbe
Router(config-sbc-sbe)# enum 1
Router(config-sbc-sbe-sbe)# req-timeout 10000
Router(config-sbc-sbe-sbe)# max-recursive-depth 100
Router(config-sbc-sbe-sbe)# entry ENUM_1
Router(config-sbc-sbe-sbe-sbe)# server ipv4 10.10.10.10 vrf VRF1
Router(config-sbc-sbe-sbe-sbe-sbe)# dial-plan-suffix Example.Suffix
Router(config-sbc-sbe-sbe-sbe-sbe)# max-responses 100
Router(config-sbc-sbe-sbe-sbe-sbe)# activate
Router(config-sbc-sbe-sbe-sbe-sbe)# end
```

Example:  
```
Router(config-sbc-sbe-sbe-sbe-sbe)# action complete
```

Step 41  
**action complete**  
Configures the action to take on routing table entry 3. In this case, the action is complete.

Example:  
```
Router(config-sbc-sbe-sbe-sbe-sbe)# action complete
```

Step 42  
**complete**  
Completes the call-policy set after committing the full set.

Example:  
```
Router(config-sbc-sbe-sbe-sbe-sbe)# complete
```

Step 43  
**end**  
Exits configuration mode and returns to privileged EXEC mode.

Example:  
```
Router(config-sbc-sbe-sbe-sbe-sbe)# end
```

Example 2: Call Policy for Multiple ENUM Entries

Use the following procedure to configure a call policy for multiple ENUM entries:

```
Router# configure terminal
Router(config)# sbc mysbc
Router(config-sbc)# sbe
Router(config-sbc-sbe)# enum 1
Router(config-sbc-sbe-sbe)# entry default-enum
Router(config-sbc-sbe-sbe-sbe)# server ipv4 192.168.10.1
Router(config-sbc-sbe-sbe-sbe-sbe)# dial-plan-suffix e164.arpa
Router(config-sbc-sbe-sbe-sbe-sbe)# entry cisco-enum
Router(config-sbc-sbe-sbe-sbe-sbe)# server ipv4 10.0.0.22 vrf cisco-vrf
Router(config-sbc-sbe-sbe-sbe-sbe-sbe)# dial-plan-suffix cisco.com
Router(config-sbc-sbe-sbe-sbe-sbe-sbe)# activate
Router(config-sbc-sbe-sbe-sbe-sbe-sbe)# exit
```

Example:  
```
Router(config-sbc-sbe-sbe-sbe-sbe-sbe)# action complete
```

Step 41  
Configures the action to take on routing table entry 3. In this case, the action is complete.

Example:  
```
Router(config-sbc-sbe-sbe-sbe-sbe-sbe)# action complete
```

Step 42  
**complete**  
Completes the call-policy set after committing the full set.

Example:  
```
Router(config-sbc-sbe-sbe-sbe-sbe-sbe)# complete
```

Step 43  
**end**  
Exits configuration mode and returns to privileged EXEC mode.

Example:  
```
Router(config-sbc-sbe-sbe-sbe-sbe-sbe)# end
```
Chapter 53  ENUM Client

**Configuration Examples of ENUM Client Configuration**

Router(config-sbc-sbe-sip-hdr-dst)# header-prio 2 header-name Dst_Add_Hdr_2
Router(config-sbc-sbe-sip-hdr-dst)# exit
or
Router(config-sbc-sbe)# sip header-profile enum
Router(config-sbc-sbe-sip-hdr)# src-address
Router(config-sbc-sbe-sip-hdr-src)# header-prio 1 header-name Src_Add_Hdr_1
Router(config-sbc-sbe-sip-hdr-src)# header-prio 2 header-name Src_Add_Hdr_2
Router(config-sbc-sbe-sip-hdr-src)# exit
or
Router(config-sbc-sbe)# sip header-profile enum
Router(config-sbc-sbe-sip-hdr)# div-address
Router(config-sbc-sbe-sip-hdr-div)# header-prio 1 header-name Div_Add_Hdr_1
Router(config-sbc-sbe-sip-hdr-div)# header-prio 2 header-name Div_Add_Hdr_2
Router(config-sbc-sbe-sip-hdr-div)# exit

Router(config-sbc-sbe-sip-hdr)# call-policy-set 1
Router(config-sbc-sbe-rtgpolicy)# first-call-routing-table rtl
Router(config-sbc-sbe-rtgpolicy)# rtg-src-adjacency-table rtl

Router(config-sbc-sbe-rtgpolicy-rtgtable)# entry 1
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# match-adjacency sip2
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# enum 1 entry default-enum
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# action next-table dal

Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# entry 2
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# match-adjacency sip2
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# enum 1 entry cisco-enum
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# dst-adjacency sip-proxy1
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# action complete
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# rtg-dst-address-table dal

Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# entry 1
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# match-address bob
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# dst-adjacency sip-proxy2
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# action complete

Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# entry 2
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# match-address kate
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# dst-adjacency sip-proxy3
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# action complete

Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# entry 3
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# match-address 44
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# prefix
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# dst-adjacency sip-proxy4
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# action complete
Router(config-sbc-sbe-rtgpolicy-rtgtable-entry)# complete
Router(config-sbc-sbe-rtgpolicy)# end
Router#
IPv6 Support

Cisco Unified Border Element (SP Edition) supports IPv6 addressing on the unified model for SIP signaling and media. Cisco Unified Border Element (SP Edition) has the ability to handle IPv4 to IPv6 SIP signaling and media interworking, as well as IPv6 to IPv6 SIP signaling and media (RTP) interworking.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).

For a complete description of the commands used in this chapter, refer to the Cisco Unified Border Element (SP Edition) Command Reference: Unified Model at:


For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

Feature History for IPv6 Support

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 2.6</td>
<td>IPv6 Support features were introduced on the Cisco ASR 1000 Series Router.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 3.1S</td>
<td>IPv6 Support for VRF was added on the Cisco ASR 1000 Series Router.</td>
</tr>
</tbody>
</table>

Contents

This module contains the following sections:

- Prerequisites, page 54-2
- Restrictions, page 54-2
- Information About IPv6 Support, page 54-2
- Configuring IPv6, page 54-5
Prerequisites

The following prerequisite is required to implement IPv6 Support:
Before implementing IPv6 Support, Cisco Unified Border Element (SP Edition) must already be configured.

Restrictions

The following are restrictions for IPv6 support on the Cisco Unified Border Element (SP Edition):
- H.323 over IPv6 is not supported.
- H.248 over IPv6 is not supported.
- The SBC does not support receiving and sending multiple IP addresses per media stream.
  For more information, refer to RFC 4091, The Alternative Network Address Types (ANAT) Semantics for the Session Description Protocol (SDP) Grouping Framework and ICE (Interactive Connectivity Establishment).
- DNS look up over IPv6 is not supported.
- RADIUS (accounting and authentication) over IPv6 is not supported.

Information About IPv6 Support

In Cisco IOS XE Release 2.6, Cisco Unified Border Element (SP Edition) supports IPv6 addressing on the unified model for SIP signaling and media in the following ways:
- IPv4 to IPv6 SIP signaling interworking
- IPv4 to IPv6 media interworking
- IPv6 to IPv6 SIP signaling
- IPv6 to IPv6 media RTP interworking
- AAAA DNS query support

IPv6 to IPv6 RTP interworking on the media plane have been supported on the distributed model. The unified model now can enable IPv6 to IPv6 SIP signaling calls and IPv4 to IPv6 SIP signaling and media interworking calls.

The default behavior is that SBC assumes that the media address type to be used must match the signaling address type configured on the adjacency. Thus the default behavior which can be overridden by configuring a Call Admission Control (CAC) policy is for the media (RTP) to use the same version as used by signaling (SIP). The IP version used by SIP is dictated by the IP addresses configured on the adjacency. For example, if the incoming SIP INVITE comes in on an IPv4 adjacency and is routed out via an IPv6 adjacency, the incoming RTP will come over IPv4 and will be sent out over IPv4.

IPv6 support for SIP calls affects the following SBC functions and existing unified SBC features:
- SIP URIs—IPv6 addresses are parsed in SIP URIs.
- Interworking IPv4 and IPv6 Adjacencies:
  - IPv6 addresses are passed through without modification in the Contact Username Passthrough feature.
SBC supports IP/fully-qualified domain name (FQDN) entries for IPv6 adjacencies.

**Note**
An adjacency can be configured for either IPv4 or IPv6 addresses only. Combinations of IPv4 and IPv6 addresses on the same adjacency are not supported.

**Note**
An adjacency configured for IPv4 cannot be changed to IPv6 and vice-versa, without deleting and recreating the adjacency. If the adjacency is referred to in the routing or CAC tables, these references must be removed before unconfiguring the adjacency.

- TLS over IPv6—Handles IPv6 addresses for adjacencies configured for SIP over Transport Layer Security (TLS) encryption.
- Access Authentication for SIP over IPv6
  Supports IPv6 adjacencies for SIP inbound authentication to challenge inbound SIP requests. For an incoming call over an IPv6 adjacency if the adjacency is configured for access authentication or inbound authentication, the call is challenged with a nonce (similar to what occurs in IPv4 addressing). The subsequent REGISTER message must have authentication parameters that should result in a RADIUS Access Request or an Access Accept (or reject) message. Note that the communication with the RADIUS server occurs over IPv4.
- Billing for IPv4 and IPv6 Calls
  Packetcable billing records do not have IP addresses embedded in them. Therefore, billing for IPv6 calls work in the same manner as for IPv4 calls and no additional configuration is needed in billing and RADIUS configuration. However only IPv4 addressing is supported for communicating with the RADIUS server. Both authentication and accounting requests go over IPv4, even when the requests are coming over an IPv6 adjacency. The control address used as source address in RADIUS requests is IPv4 only. The billing manager local address goes in the NAS IP address field of RADIUS requests. This address is also an IPv4 address.
- SRTP Passthrough mode—No additional configuration changes for IPv6 addressing.
- Media bypass in Call Admission Control—The SBC must not attempt to perform media bypass between endpoints with different IP versions, even if media bypass CAC policy permits it.
- Blacklist Support—Supports the configuration of blacklist entries with IPv6 addresses or prefixes in the same way as IPv4 addresses.
- Logging—Displays IPv6 and IPv4 addresses.
- Late-to-Early Media Interworking—Supports calls terminating and originating from an IPv6 adjacency.
- Softswitch Shielding—Supports IPv6 endpoints registering with a softswitch.
- Call Hold—Supports IPv4 to IPv6 call interworking.
  Using “c=0.0.0.0” as specified in RFC 2543 to indicate call hold is not valid with IPv6 addresses, and you must use “a=sendonly/inactive” to indicate a call hold.
- ToS/DSCP Marking for Signaling Messages—Supports DSCP marking for outgoing IPv4 and IPv6 signaling packets.
• SIP Header Manipulation—Supports the passthrough header TO and FROM functionality for IPv6 to IPv4 interworked calls by passing the headername unchanged for incoming calls. SBC rewrites the CONTACT header for outgoing calls.

• DTMF Interworking—Supports IPv6 adjacencies.

• IP Realms—Supports IPv6 adjacencies. IP addresses are assigned based on the realm configured on the IPv6 or IPv4 adjacency.

• SIP Instant Messaging—Supports IPv4 to IPv6 interworked calls.

• SIP IP-FQDN URI Translation—Supports IP-FQDN entries for IPv6 adjacencies.

• Domain Name Lookup (DNS)—Supports name lookup for IPv6 addresses and supports both A and AAAA DNS queries. DNS lookup happens over IPv4.

• Fast Registration—Supports IPv6 addresses.

• High Availability—IPv4 to IPv6 and IPv6 to IPv6 calls behave the same as IPv4 to IPv4 calls during failover. Calls over UDP are replicated. For calls made over TCP, the signaling state is not replicated and will generate a TCP reset on receiving any SIP message after switchover.

• Media Hair-pinning—Supports IPv6 to IPv6 call hair pinning in the same manner as IPv4 to IPv4 calls. With media hair-pinning, calls come in and go back out on the same adjacency.

Note IPv4 to IPv6 hair-pinning is not supported because an adjacency can only be an IPv4 adjacency or an IPv6 adjacency.

• 3xx Redirect Messages—Supports redirection from IPv4 to IPv6 and from IPv6 to IPv4.

3xx represents a class of SIP response codes used in SIP to indicate that the sender of the request should try the request to an alternate URI or URIs that are presented in the 3xx response. Some of the widely-used response code examples are 301 “Moved Temporarily” or 302 “Moved Permanently.”

Performing ISSU for IPv6 Calls

When performing ISSU to upgrade to a higher version Cisco IOS XE release, IPv4 to IPv4 calls migrate successfully to the higher version.

Before performing ISSU to migrate to a lower version release, you must first unconfigure all IPv6 adjacencies and remove all active IPv6 call states. You can clear calls through IPv6 adjacencies with the no attach force abort command. This command executes a forced detach, tearing down calls without signaling their end.

When performing ISSU to downgrade to a lower version Cisco IOS XE release, for example from Cisco IOS XE Release 2.6 to 2.5, if there is any IPv6 configuration or any active calls through an IPv6 adjacency, an error message is reported. If the user continues with the ISSU, the system will reach stateful switchover (SSO) without the SBC configuration being available on the standby processor. Before performing a downgrade, unconfigure all IPv6 configuration and dynamic state (for example, IPv6 to IPv6 and IPv6 to IPv4 calls, as well as IPv6 blacklists).
Configuring IPv6

To configure Cisco Unified Border Element (SP Edition) for IPv6 to IPv6 calls or IPv4 to IPv6 interworked calls, configure the local and remote addresses on the adjacency with IPv6 addresses.

If you have a peer or another SBC in your network that supports both IPv4 and IPv6 addresses, then you should define two adjacencies on the local SBC, one adjacency with IPv4 addresses and a second adjacency with IPv6 addresses.

Configuration Examples

The following example shows the asr1 SBC configured with IPv6 and IPv4 signaling and remote addresses on several SIP adjacencies and the 1 call policy set using a round-robin routing rule to implement call routing:

```
! sbc asr1
sbe
  control address aaa ipv4 33.33.36.1
  radius authentication
  radius accounting server1
  server server1
    address ipv4 10.0.120.19
    key cisco
  activate
  sip header-profile ccmpf1
  header Allow entry 1
    action pass
  header Call-Info entry 1
    action pass
  sip method-profile 1
    pass-body
    method MESSAGE
    action pass
  sip method-profile method1
    pass-body
    method INFO
    action pass
  sip method-profile ccmmethod1
    pass-body
    method SUBSCRIBER
    action pass
  sip method-profile ccmmethod2
    pass-body
    method INFO
    action pass
  method NOTIFY
    action pass
  method SUBSCRIBER
    action pass
  adjacency sip UEV6
```
group IPv6
inherit profile preset-p-cscf-access
visited network identifier open-ims.test
local-id host pcsfc.open-ims.test
signaling-address ipv6 2001:AA01::33:33:36:1
statistics method summary
signaling-port 4060
remote-address ipv6 2001::/64
signaling-peer 2001::10:0:120:19
attach
adjacency sip CCM134
force-signaling-peer
group v4
nat force-on
header-profile inbound ccmpf1
header-profile outbound ccmpf1
method-profile inbound cccmethod2
method-profile outbound cccmethod2
preferred-transport udp
signaling-address ipv4 33.33.36.1
statistics method summary
signaling-port 5060
remote-address ipv4 10.0.50.134 255.255.255.255
signaling-peer 10.0.50.134
dbe-location-id 0
account CCM134
media-late-to-early-iw incoming
media-late-to-early-iw outgoing
dtmf disable sip notify
dtmf prefer sip info
attach
adjacency sip CCM135
group v4
nat force-on
header-profile inbound ccmpf1
header-profile outbound ccmpf1
preferred-transport udp
signaling-address ipv4 33.33.36.1
statistics method summary
signaling-port 5060
remote-address ipv4 10.0.50.135 255.255.255.255
signaling-peer 10.0.50.135
dbe-location-id 0
attach
adjacency sip CCM136
force-signaling-peer
redirect-mode recurse
signaling-address ipv4 33.33.36.1
statistics method summary
signaling-port 5060
remote-address ipv4 10.0.50.136 255.255.255.255
signaling-peer 10.0.50.136
dbe-location-id 0
ping-enable
ping-interval 60
ping-lifetime 2
attach
adjacency sip CSPS23
nat force-off
preferred-transport udp
signaling-address ipv4 33.33.36.1
statistics method summary
remote-address ipv4 10.0.7.23 255.255.255.255
Chapter 54  IPv6 Support

Configuring IPv6

```
signaling-peer 10.0.7.23
dbe-location-id 0
attach
adjacency sip OpensipsV6
group IPv6
nat force-off
inherit profile preset-core
    signaling-address ipv6 2001:A401::33:33:36:1
statistics method summary
signaling-port 7060
remote-address ipv6 2001::216:ECFF:FE3B:40DD/128
signaling-peer opensips.cisco.com
dbe-location-id 0
registration target address opensips.cisco.com
header-name From passthrough
dtmf prefer sip info
attach
adjacency sip CCM135-IPv6
force-signaling-peer
group v6
nat force-off
header-profile inbound ccmpf1
header-profile outbound ccmpf1
method-profile inbound ccmmethod2
method-profile outbound ccmmethod2
preferred-transport udp
    signaling-address ipv6 2001:A401::33:33:36:1
statistics method summary
signaling-port 5060
remote-address ipv6 2001::10:0:50:135/128
signaling-peer 2001::10:0:50:135
dbe-location-id 0
attach
adjacency sip CCM135-vrfb
vrf h323-vrf-b
nat force-off
preferred-transport udp
    signaling-address ipv4 10.190.7.97
statistics method summary
signaling-port 5060
remote-address ipv4 10.0.50.135 255.255.255.255
signaling-peer 10.0.50.135
dbe-location-id 0
attach
adjacency sip CCM136-IPv6
group v6
nat force-off
header-profile inbound ccmpf1
header-profile outbound ccmpf1
method-profile inbound ccmmethod2
method-profile outbound ccmmethod2
    signaling-address ipv6 2001:A401::33:33:36:1
statistics method summary
signaling-port 5060
remote-address ipv6 2001::10:0:50:136/128
signaling-peer 2001::10:0:50:136
ping-enable
    ping-interval 60
    ping-lifetime 2
dtmf prefer sip info
attach
adjacency sip SIPP81-IPv6
group v6
nat force-off
```
preferred-transport udp

signaling-address ipv6 2001:A401::33:33:36:1
statistics method summary
signaling-port 5060

remote-address ipv6 2001::/64
signaling-peer 2001::10:0:244:81
dbe-location-id 0
dtmf disable sip notify
dtmf prefer sip info
attach
call-policy-set 1
first-call-routing-table ROUTE-ON-DEST-NUM
first-reg-routing-table REG-ROUTE-ON-SRC-ADJ
rtg-src-adjacency-table REG-ROUTE-ON-SRC-ADJ
entry 1
  action complete
dst-adjacency OpensipsV6
match-adjacency UEV6
rtg-round-robin-table ROUND-ROBIN
entry 1
  action complete
dst-adjacency CCM136
entry 2
  action complete
dst-adjacency CCM136-IPv6
rtg-dst-address-table ROUTE-ON-DEST-NUM
entry 1
  action next-table ROUND-ROBIN
edit del-prefix 3
match-address 536X digits
prefix
entry 2
  action next-table ROUND-ROBIN
edit del-prefix 4
match-address 7898X digits
prefix
entry 3
  action next-table ROUND-ROBIN
edit del-prefix 3
match-address 491X digits
prefix
entry 4
  action next-table ROUND-ROBIN
edit del-prefix 3
match-address 526X digits
prefix
entry 5
  action next-table ROUND-ROBIN
edit del-prefix 3
match-address 496X digits
prefix
entry 6
  action complete
edit del-prefix 3
dst-adjacency CCM135
match-address 4553X digits
prefix
entry 7
  action complete
edit del-prefix 3
dst-adjacency CCM135
match-address 789X digits
prefix
entry 8
The following example shows different signaling and media addresses configured on the SBC asr1, where the signaling address configured is ipv6 and the media address configured is ipv4:
sbc asr1
sbe
  adjacency sip CCM1-IPV6
group media-v4
  nat force-off
  preferred-transport udp
  signaling-address ipv6 2001:A401::33:33:36:1
  statistics method summary
  signaling-port 5060
  remote-address ipv6 2001::10:0:56:186/128
  signaling-peer 2001::10:0:56:186
  dbe-location-id 0
  attach
  adjacency sip CCM2-IPV6
group media-v4
  nat force-off
  preferred-transport udp
  signaling-address ipv6 2001:A401::33:33:36:1
  statistics method summary
  signaling-port 5060
  remote-address ipv6 2009::100:0:0:4/128
  signaling-peer 2009::100:0:0:4
  dbe-location-id 0
  attach
cac-policy-set 1
  first-cac-table table1
cac-table table1
  table-type limit account
  entry 1
  match-value media-v4
  action cac-complete
    caller media-type ipv4
    callee media-type ipv4
    complete
active-cac-policy-set 1
call-policy-set 1
first-call-routing-table table1
rtg-dst-address-table table1
entry 1
  action complete
  edit del-prefix 3
dst-adjacency CCM2-IPV6
match-address 123X digits
prefix
  complete
active-call-policy-set 1
!
!
media-address ipv4 33.33.36.10
media-timeout 360
activate
!
!
IPv6 Configuration Commands

This section describes the configuration commands used to configure various types of IPv6 addressing or show output listing IPv6 addresses.


Table 54-1 lists the new commands introduced in Cisco IOS XE Release 2.6.

Table 54-1  New Commands Introduced in Cisco IOS XE Release 2.6

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>callee media-type {ipv4</td>
<td>ipv6</td>
</tr>
<tr>
<td>caller media-type {ipv4</td>
<td>ipv6</td>
</tr>
</tbody>
</table>

Table 54-2 lists the commands modified for IPv6 addressing in Cisco IOS XE Release 2.6.

Table 54-2  Commands Modified for IPv6 Addressing in Cisco IOS XE Release 2.6

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>blacklist [critical] global [address-default</td>
<td>{ipv4 (addr)</td>
</tr>
<tr>
<td>clear sbc sbc-name sbc blacklist [ critical ] {ipv4 addr</td>
<td>ipv6 addr} [{udp</td>
</tr>
<tr>
<td>remote-address {ipv4 ip-address ip-mask</td>
<td>ipv6 ip-address / prefix-length}</td>
</tr>
<tr>
<td>show sbc sbc-name sbc addresses</td>
<td>The output of this command was modified.</td>
</tr>
<tr>
<td>show sbc sbc-name sbc adjacencies {adjacency-name} [detail]</td>
<td>The output of this command was modified.</td>
</tr>
</tbody>
</table>
### Table 54-2  Commands Modified for IPv6 Addressing in Cisco IOS XE Release 2.6

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>show sbc sbc-name sbe blacklist critical { ipv4 addr \ ipv6 addr } [ tcp tcp-port</td>
<td>udp udp-port ]</td>
</tr>
<tr>
<td>show sbc sbc-name sbe blacklist {source} { ipv4 IP address \ ipv6 IP address }</td>
<td>The <strong>ipv6</strong> keyword was added.</td>
</tr>
<tr>
<td>show sbc name sbc cac-policy-set { id / table name [ entry id ] }</td>
<td>active / table name [ entry id ] [ detail ]</td>
</tr>
<tr>
<td>show sbc sbc-name sbe calls</td>
<td>To provide details of IPv6 calls.</td>
</tr>
<tr>
<td>show sbc sbc-name sbe call-stats { all</td>
<td>global</td>
</tr>
<tr>
<td>show sbc sbc-name sbe addresses</td>
<td>The output of this command was modified.</td>
</tr>
<tr>
<td>show sbc sbc-name sbe sip ip-fqdn-mapping</td>
<td>Displays the IP-FQDN mapping table. The output of this command was modified to include IPv6 details.</td>
</tr>
<tr>
<td>signaling-address {ipv4 ipv4_IP_address</td>
<td>ipv6 ipv6_IP_address}</td>
</tr>
<tr>
<td>sip ip-fqdn-mapping index { ipv4</td>
<td>ipv6 } ip-address fqdn { both-ways</td>
</tr>
</tbody>
</table>

Table 54-3 lists the command modified for IPv6 addressing in Cisco IOS XE Release 3.1.0S.

### Table 54-3  Command Modified for IPv6 Addressing in Cisco IOS XE Release 3.1.0S

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>blacklist [critical] vpn { vpn-name } [ address-default { address-family { ipv4</td>
<td>ipv6 } ] [ address-family { ipv4</td>
</tr>
</tbody>
</table>
Table 54-4 lists the command modified for IPv6 addressing in Cisco IOS XE Release 3.5.0S.

Table 54-4 Command Modified for IPv6 Addressing in Cisco IOS XE Release 3.5.0S

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| branch media-type {ipv4 | ipv6 | inherit | both} | Configures the media address type settings for a caller or callee on the Cisco Unified Border Element (SP Edition).
  
  Use the no form of this command to disable the media address type settings for the caller or callee. |
P-CSCF Support

The Proxy-Call Session Control Function (P-CSCF) is the first contact point for the users of the IP Multimedia Subsystem (IMS). The P-CSCF functions as a proxy server for the user equipment; all Session Initiation Protocol (SIP) signaling traffic to and from the user equipment must go through the P-CSCF. The P-CSCF validates and then forwards requests from the user equipment and then processes and forwards the responses to the user equipment.

The P-CSCF can also function as a user agent in the context of the SIP operating procedures. If an abnormal condition arises in a session, the P-CSCF can unilaterally release the session for the user equipment. The user agent role can also be used to generate independent SIP messages required during the registration, such as sending the user’s public and private identities. There may be more than one P-CSCF in the operator’s network based on survivability, number of users, expected traffic, and network topology. The P-CSCF can be also referred to as the SIP server.

To implement the P-CSCF support on Cisco Unified Border Element (SP Edition), users must select an Inherit Profile for a SIP adjacency. The three available Inherit Profiles are:

- Standard Non-IMS Profile
- P-CSCF Access Profile
- P-CSCF Core Profile

Each of these profiles groups a set of IMS-related configuration fields that can be applied across multiple adjacencies.

If a valid profile is configured, this profile is applied to an adjacency that does not have a profile configured. If a profile is already selected for a SIP adjacency, that profile is used instead of the entity’s profile.

In Cisco IOS XE Release 2.5 and later, Cisco Unified Border Element (SP Edition) supports Hypertext Transfer Protocol (HTTP) Digest Authentication Using Authentication and Key Agreement (AKA) for SIP calls. This type of authentication is used for access authentication in mobile IMS deployments and typically may reside on a mobile subscriber’s card inside a phone. No special configuration is needed. The only requirement is that a UNI SIP profile is configured on the access side of the network.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).

For a complete description of the commands used in this chapter, refer to the Cisco Unified Border Element (SP Edition) Command Reference: Unified Model at:


For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.
For Cisco IOS XR Software Release and later, this feature is supported in the unified model only.

Feature History for P-CSCF Support

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XR Software Release</td>
<td>This support was introduced on the Cisco IOS XR along with support for the unified model.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 2.5</td>
<td>The HTTP Digest Authentication Using AKA feature was introduced on the Cisco ASR 1000 Series Routers.</td>
</tr>
</tbody>
</table>

Contents

This module contains the following sections:

- Restrictions for Implementing P-CSCF Support, page 55-2
- Information About P-CSCF Support, page 55-2
- Implementing P-CSCF Support, page 55-6
- Information About HTTP Digest Authentication Using AKA, page 55-7

Restrictions for Implementing P-CSCF Support

The following restrictions and limitations apply to implementing P-CSCF support:

- Since the Visited Network Identifier is not part of an Inherit Profile, you need to configure it independently on a per-adjacency basis.
- This feature does not offer support for securing access links through IPsec or Network Attachment Subsystem (NASS) bundled authentication.
- This feature does not support emergency calls.

Information About P-CSCF Support

This section contains the following subsections:

- Standard Non-IMS Profile, page 55-3
- P-CSCF Access Profile, page 55-3
- P-CSCF Core Profile, page 55-3
- Effect of P-CSCF Inherit Profiles on Method Profiles, Header Profiles, and Option Profiles, page 55-4
Standard Non-IMS Profile

This profile provides compatibility with existing Cisco Unified Border Element (SP Edition) functionality and is used for adjacencies that do not operate in an IMS network. When this profile is applied to an adjacency, Cisco Unified Border Element (SP Edition) exhibits the following properties:

- Contact headers are rewritten to ensure that the SBC remains on the signaling path.
- Unknown headers, methods, and options are, by default, not allowed to pass through.
- Cisco Unified Border Element (SP Edition) does not attach Path headers to outbound signals.
- Cisco Unified Border Element (SP Edition) does not attach Record-Route headers to outbound signals.
- The endpoints on this adjacency do not need to be registered to receive or send Non-REGISTER requests.
- The endpoints do not need to attach a Route header to outbound signals.
- The adjacencies do not generate P-Charging Vector headers for outbound signals.

P-CSCF Access Profile

This profile provides the configurations required to perform the functions of a P-CSCF Access adjacency. When this profile is applied to an adjacency, Cisco Unified Border Element (SP Edition) exhibits the following properties:

- Contact headers are not rewritten.
- The endpoints on this adjacency need to be registered to receive or send Non-REGISTER requests.
- The endpoints need to attach a Route header to outbound signals, which in turn, matches a Service-Route set from the Registrar.
- The SBC appends Record-Route headers to outbound signals for adjacencies with P-CSCF profiles.
- The SBC does not attach Path headers to outbound signals.
- The adjacencies do not generate P-Charging Vector headers for outbound signals.
- The SBC, by default, allows all outbound non-essential headers, except P-Charging-Function-Addresses, P-Charging-Vector, and P-Media-Authorization.
- The SBC allows all inbound non-essential methods to pass through.
- The SBC allows all outbound non-essential methods to pass through; UEs are not permitted to act as Registrars.
- The Option tags in Supported, Require, or Proxy-Require headers are allowed to pass through in both directions.

P-CSCF Core Profile

This profile provides the configurations required to perform the functions of a P-CSCF Core adjacency. When this profile is applied to an adjacency, Cisco Unified Border Element (SP Edition) exhibits the following properties:
Information About P-CSCF Support

- Contact headers are not rewritten.
- The SBC, by default, allows all inbound unknown headers, except the P-Charging-Function-Addresses and P-Media-Authorization.
- The SBC appends Record-Route headers to outbound signals for adjacencies with P-CSCF profiles.
- The SBC attaches Path headers to outbound REGISTER signals from P-CSCF.
- The adjacencies generate P-Charging Vector headers for outbound signals.
- The endpoints on this adjacency do not need to be registered to receive or send Non-REGISTER requests.
- The SBC, by default, allows all outbound non-essential headers, except P-Charging-Function-Addresses and P-Media-Authorization.
- The SBC allows all unknown methods to pass through.
- The Option tags in Supported, Require, or Proxy-Require headers are allowed to pass through in both directions.

Effect of P-CSCF Inherit Profiles on Method Profiles, Header Profiles, and Option Profiles

Use of a P-CSCF inherit profile dynamically assigns the following sets of profiles (method profile, header profile, and option profile) to a call based on the P-CSCF inherit profile selected. Table 55-1 shows which P-CSCF inherit profile has an effect on which specific method profile, header profile, and option profile.

The effect is not visible in the adjacency configuration for header-profile, method-profile or option profiles, and can be overridden by explicit configuration of header, method, option profiles as needed.
### Table 55-1  Effect of P-CSCF Inherit Profiles on Method, Header and Option Profiles

<table>
<thead>
<tr>
<th>P-CSCF Inherit Profile</th>
<th>Method Profile</th>
<th>Header Profile</th>
<th>Option Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td>preset-p-cscf-access</td>
<td>preset-acc-in-mth</td>
<td>preset-acc-in-hdr</td>
<td>preset-acc-in-opt</td>
</tr>
<tr>
<td></td>
<td>Type: Blacklist</td>
<td>Type: Blacklist</td>
<td>Type: Blacklist</td>
</tr>
<tr>
<td></td>
<td>Actions: No methods rejected</td>
<td>Actions: Removes Security-Client</td>
<td>Actions: No options (Passes on all)</td>
</tr>
<tr>
<td></td>
<td>preset-acc-out-mth</td>
<td>preset-acc-out-hdr</td>
<td>preset-acc-out-opt</td>
</tr>
<tr>
<td></td>
<td>Type: Blacklist</td>
<td>Type: Blacklist</td>
<td>Type: Blacklist</td>
</tr>
<tr>
<td></td>
<td>Actions: Removes P-Charging-Vector</td>
<td>Actions:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>preset-acc-in-opt</td>
<td>no options</td>
<td></td>
</tr>
<tr>
<td></td>
<td>preset-acc-out-opt</td>
<td>(Passes on all)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Type: Blacklist</td>
<td>Type: Blacklist</td>
<td>Type: Blacklist</td>
</tr>
<tr>
<td></td>
<td>Actions: No methods removed</td>
<td>Actions: Removes no headers (passes all)</td>
<td>Actions: No options (Passes on all)</td>
</tr>
<tr>
<td></td>
<td>preset-core-out-mth</td>
<td>preset-core-out-hdr</td>
<td>preset-core-out-opt</td>
</tr>
<tr>
<td></td>
<td>Type: Blacklist</td>
<td>Type: Blacklist</td>
<td>Type: Blacklist</td>
</tr>
<tr>
<td></td>
<td>Actions: Removes P-Charging-Vector</td>
<td>Actions:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>preset-core-in-opt</td>
<td>no options</td>
<td></td>
</tr>
<tr>
<td></td>
<td>preset-core-out-opt</td>
<td>(Passes on all)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>preset-std-out-mth</td>
<td>preset-std-out-hdr</td>
<td>preset-std-out-opt</td>
</tr>
<tr>
<td></td>
<td>Type: Whitelist</td>
<td>Type: Whitelist</td>
<td>Type: Whitelist</td>
</tr>
<tr>
<td></td>
<td>Actions: Passes INFO</td>
<td>Actions: Passes Server</td>
<td>Actions: Passes Replaces (only)</td>
</tr>
<tr>
<td></td>
<td>Passes UPDATE</td>
<td>Passes Diversion</td>
<td>Passes Resource-Priority</td>
</tr>
</tbody>
</table>
Implementing P-CSCF Support

This section explains how to configure intrinsic profiles and profile inheritance.

Configuring Profile Inheritance

SUMMARY STEPS

1. configure terminal
2. sbc service-name
3. sbe
4. sip inherit profile preset-p-cscf-access
5. adjacency sip adjacency-name
6. inherit profile preset-p-cscf-access
7. visited network identifier network-name
8. exit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc service-name</td>
<td>Enters the mode of an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# sbc mysbc</td>
<td>• Use the service-name argument to define the name of the service.</td>
</tr>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Enters the mode of a SBE entity within an SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> sip inherit profile preset-p-cscf-access</td>
<td>Configures the P-CSCF Access Inherit Profile as the global profile. For a list of other configurable parameters, see the sip inherit profile command.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe)# sip inherit profile preset-p-cscf-access</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> adjacency sip adjacency-name</td>
<td>Enters the mode of an SBE SIP adjacency.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-sbc-sbe)# adjacency sip sipadj</td>
<td>• Use the adjacency-name argument to define the name of the SIP adjacency.</td>
</tr>
</tbody>
</table>
Cisco Unified Border Element (SP Edition) supports Hypertext Transfer Protocol (HTTP) Digest Authentication Using Authentication and Key Agreement (AKA) for SIP calls. This type of authentication is used for access authentication in mobile IMS deployments and typically resides on a mobile subscriber’s card inside a phone. Cisco Unified Border Element (SP Edition) supports the HTTP Digest Authentication Using AKA feature with no special configuration needed, as long as a User-to-Network Interconnections (UNI) SIP profile is configured on the access side (that is, with a P-CSCF access side profile).

The AKA function carries out user authentication and session key distribution in Universal Mobile Telecommunications System (UMTS) networks. AKA is challenge-response based. The response to the challenge is computed by the application running on the mobile subscriber’s card inside the phone. Cisco Unified Border Element (SP Edition) supports the HTTP Digest Authentication Using AKA feature with no special configuration needed, as long as a User-to-Network Interconnections (UNI) SIP profile is configured on the access side (that is, with a P-CSCF access side profile).

HTTP Digest Authentication is common with IP-PBXs. The HTTP Digest Authentication procedure is used to ensure that only valid devices can register (at a SIP level) to a network. The SBC supports the typical registration call flow, that is, passing through authentication challenges and their responses. A typical call flow consists of a SIP REGISTER message from an endpoint that is routed by the SBC to the SIP registrar. The registrar replies with a 401 Unauthorized response and a “challenge.” The challenge contains a random number that the endpoint uses to compute a response, which is sent in another REGISTER message. Finally the registrar replies with a 200 OK message if the response was valid. In the case of HTTP Digest Authentication Using AKA, the response to the challenge is computed by the application running on the mobile subscriber’s card inside the phone. The SBC supports this typical call flow by means of enabling a SIP profile that allows SIP registrations.

Another usage of HTTP Digest Authentication Using AKA concerns the ability of using the procedure to establish an IPsec connection (actually two IPsec connections) for ensuring signaling security. Cisco Unified Border Element (SP Edition) supports IPsec, however the ability to extract the port security association identifiers and key information from SIP messages is not supported in Cisco IOS XE Release 2.5.

### Command or Action

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 6 inherit profile preset-p-cscf-access</td>
<td>Configures the SIP adjacency to use the P-CSCF-Access profile.</td>
</tr>
<tr>
<td>Example: Router(config-sbc-sbe-adj-sip)# inherit profile preset-p-cscf-access</td>
<td></td>
</tr>
<tr>
<td>Step 7 visited network identifier network-name</td>
<td>Configures the specified visited network identifier on the SIP adjacency.</td>
</tr>
<tr>
<td>Example: Router(config-sbc-sbe-adj-sip)# visited network identifier mynetwork.com</td>
<td></td>
</tr>
<tr>
<td>Step 8 exit</td>
<td>Exits the SIP adjacency mode to the SBE mode.</td>
</tr>
<tr>
<td>Example: Router(config-sbc-sbe-adj-sip)# exit</td>
<td></td>
</tr>
</tbody>
</table>
Configuring HTTP Digest Authentication Using AKA

This task configures HTTP Digest Authentication Using AKA on two related adjacencies where preset-access and preset-core profiles must be configured.

SUMMARY STEPS

1. configure terminal
2. sbc sbc-name
3. sbe
4. adjacency {sip | h323} adjacency-name
5. inherit profile {preset-access | preset-core | preset-ibcf-ext-untrusted | preset-ibcf-external | preset-ibcf-internal | preset-p-cscf-access | preset-p-cscf-core | preset-peering | preset-standard-non-ims}
6. exit
7. adjacency {sip | h323} adjacency-name
8. inherit profile {preset-access | preset-core | preset-ibcf-ext-untrusted | preset-ibcf-external | preset-ibcf-internal | preset-p-cscf-access | preset-p-cscf-core | preset-peering | preset-standard-non-ims}
9. exit
10. end
11. show sbc sbc-name sbe adjacencies adjacency-name detail

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enables global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc sbc-name</td>
<td>Creates the SBC service on the SBC and enters into SBC configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# sbc mySbc</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Enters the mode of the signaling border element (SBE) function of the SBC.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> adjacency {sip</td>
<td>h323} adjacency-name</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe)# adjacency sip sipEndpoint</td>
<td></td>
</tr>
</tbody>
</table>
### Information About HTTP Digest Authentication Using AKA

**Command or Action**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 5</strong></td>
<td>`inherit profile {preset-access</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td><code>exit</code></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>`adjacency {sip</td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>`inherit profile {preset-access</td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td><code>exit</code></td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td><code>end</code></td>
</tr>
<tr>
<td><strong>Step 11</strong></td>
<td><code>show sbc sbc-name sbe adjacencies adjacency-name detail</code></td>
</tr>
</tbody>
</table>

**Purpose**

- **Step 5:** Required. Configures a preset P-CSCF access profile for the SIP adjacency facing the endpoint. P-CSCF is Proxy-Call Session Control Function—part of its function is to authenticate the user and establish an IPsec security association with the IMS terminal.
- **Step 6:** Exits adjacency sip configuration mode and enters into sbe configuration mode.
- **Step 7:** Configures the SIP adjacency facing the registrar/softswitch, and enters into adjacency sip configuration mode.
- **Step 8:** Required. Configures a preset P-CSCF core profile for the SIP adjacency facing the registrar/softswitch. An adjacency facing the registrar typically has a preset-core profile. The default is preset-core.
- **Step 9:** Exits adjacency sip configuration mode and enters into SBE configuration mode.
- **Step 10:** Exits SBE configuration mode and returns to EXEC mode.
- **Step 11:** Displays all the detailed field output for the specified SIP adjacency.

**Example:**

- `Router(config-sbc-sbe-adj-sip)# inherit profile preset-p-cscf-access`
- `Router(config-sbc-sbe-adj-sip)# exit`
- `Router(config-sbc-sbe)# adjacency sip SoftSwitch`
- `Router(config-sbc-sbe-adj-sip)# inherit profile preset-p-cscf-core`
- `Router(config-sbc-sbe-adj-sip)# exit`
- `Router(config-sbc-sbe)# end`
- `Router(config-sbc-sbe-adj-sip)# show sbc sbc-name sbe adjacencies adjacency-name detail`
Configuration Example—HTTP Digest Authentication Using AKA

The following is a configuration example used to verify HTTP Digest Authentication Using AKA:

```
sbc asr
  sbe
    adjacency sip UE
      inherit profile preset-p-cscf-access
      visited network identifier open-ims.test
      local-id host pcscf.open-ims.test
      signaling-address ipv4 10.190.5.129
      signaling-port 4060
      remote-address ipv4 10.0.0.0 255.255.0.0
      signaling-peer 10.0.120.19
      dbe-location-id 100
      fast-register disable
      attach

  adjacency sip OpenIMSCore
    inherit profile preset-p-cscf-core
    visited network identifier open-ims.test
    local-id host pcscf.open-ims.test
    signaling-address ipv4 10.190.5.129
    signaling-port 4060
    remote-address ipv4 10.0.48.236 255.255.255.255
    signaling-peer 10.0.48.236
    dbe-location-id 100
    registration rewrite-register
    registration target address open-ims.test
    attach
```
IBCF Processing Support

Users can configure Cisco Unified Border Element (SP Edition) to perform the role of an Interconnection Border Control Function (IBCF) Session Initiation Protocol (SIP) border gateway, both managing requests across a network border between IP Multimedia Subsystem (IMS) core networks and interworking with non-IMS core networks.

When functioning as an IBCF, Cisco Unified Border Element (SP Edition) supports the following IBCF functions:

- Adding to Path header on REGISTER
- Modifying Service Route header
- Routing based on SIP Route headers
- Topology hiding
- Screening of SIP signaling
- IBCF inherit profiles
- Passthrough of From, To, and Contact headers
- Passthrough of request Uniform Resource Identifier (URI) on REGISTER
- Interworking with Proxy Call Session Control Function (P-CSCF), Interrogating Call Session Control Function (I-CSCF, and Serving Call Session Control Function (S-CSCF)
- Handling messages from untrusted domains
- Adding Record-Route headers on outbound messages for adjacencies with IBCF profiles.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the session border controller (SBC).

For a complete description of the commands used in this chapter, refer to the Cisco Unified Border Element (SP Edition) Command Reference: Unified Model at:


For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

Note: For Cisco IOS XE Release 2.4, this feature is supported in the unified model only.
Feature History for IBCF Support

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 2.4</td>
<td>This feature was introduced on the Cisco CRS-1 along with support for the unified model.</td>
</tr>
</tbody>
</table>

Contents

This module contains the following sections:

- Restrictions for Implementing IBCF Support, page 56-2
- Information About IBCF Support, page 56-2
- Implementing IBCF Support, page 56-5

Restrictions for Implementing IBCF Support

The following features are not included in the IBCF support:

- Blacklist or whitelist header-values-content-type, content-disposition, and content-language headers
- Blacklist or whitelist MIME bodies
- Session timer
- Co-location with I-CSCF
- Cisco Unified Border Element (SP Edition) does not reject long message bodies.
- Cisco Unified Border Element (SP Edition) does not check the length of SIP bodies.
- Cisco Unified Border Element (SP Edition) does not provide a default implementation of the Encryption User Exit.
- Cisco Unified Border Element (SP Edition) does not hide network devices that are identified by IP addresses.
- Cisco Unified Border Element (SP Edition) does not support the full IBCF handling of failed REGISTERs.
- Cisco Unified Border Element (SP Edition) does not provide interoperability between IMS and other SIP domains.
- The IBCF selection of a new entry point for forwarding REGISTER requests is limited to SIP Server Location procedures (as per IETF RFC 3263) and is applicable only if the initial server selected does not respond.

Information About IBCF Support

This section contains the following subsections:

- Adding to Path Header on REGISTER, page 56-3
- Modifying Service-Route Header on REGISTER, page 56-3
- Routing Based on SIP Route Headers, page 56-3
• Topology Hiding, page 56-3
• Screening of SIP Signaling, page 56-3
• IBCF Inherit Profiles, page 56-3
• Passthrough of From, To, and Contact Headers, page 56-5
• Passthrough of Request URI on REGISTER, page 56-5
• Interworking with P-CSCF, I-CSCF, and S-CSCF, page 56-5
• Handling Messages from Untrusted Domains, page 56-5

Adding to Path Header on REGISTER

When Cisco Unified Border Element (SP Edition) is configured to perform the role of an IBCF gateway, the IBCF adds itself to the Path header to ensure that all INVITE requests to the subscriber are routed via the IBCF.

Modifying Service-Route Header on REGISTER

The Service-Route header is analogous to the Path header, but it is used to specify the list of devices a call should traverse for calls originating from a subscriber. By default, the IBCF does not modify the Service-Route header sent on REGISTER responses. However, if topology hiding is required, then the IBCF encrypts the header elements that match its configured HomeNetworkId.

Routing Based on SIP Route Headers

You can configure Cisco Unified Border Element (SP Edition) to route Dialog-creating requests, such as INVITE, to the next hop-IP address based on the Route header, which ensures that the SIP messages go through the specified border gateways between networks and the S-CSCF that handled the User Agent (UA) REGISTER.

Topology Hiding

Cisco Unified Border Element (SP Edition) hides those parts of the routing-related headers that reveal the internal topology of the SBC network. But this feature also ensures that the headers are usable for INVITE requests and other methods.

Screening of SIP Signaling

When configured to perform the role of an IBCF gateway, Cisco Unified Border Element (SP Edition) does not screen certain SIP headers using profile whitelists and blacklists.

IBCF Inherit Profiles

IBCF inherit profiles comprise a collection of related configuration appropriate to a particular network role. IBCF Inherit profiles may be configured for an application on a per-adjacency basis.
Cisco Unified Border Element (SP Edition) supports the following IBCF inherit profiles:

- preset-ibcf-ext-untrusted
- preset-ibcf-external
- preset-ibcf-internal

Use of an IBCF inherit profile dynamically assigns a method profile, header profile, and/or option profile to a call based on the inherit-profile selected. Table 56-1 shows which IBCF inherit profile has an effect on which specific method profile, header profile, and option profile.

The effect is not visible in the adjacency configuration for header-profile, method-profile or option profiles, and can be overridden by explicit configuration of header, method, option profiles as needed.

**Table 56-1**  
Effect of IBCF Inherit Profiles on Method, Header and Option Profiles

<table>
<thead>
<tr>
<th>IBCF Inherit Profile</th>
<th>Method Profile</th>
<th>Header Profile</th>
<th>Option Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>preset-ibcf-utr-out-mth</td>
<td>Type: Blacklist Actions: No methods rejected</td>
<td>preset-ibcf-utr-out-opt</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Type: Blacklist Actions: No options (passes on all)</td>
</tr>
<tr>
<td></td>
<td>preset-ibcf-ext-out-mth</td>
<td>Type: Blacklist Actions: No methods rejected</td>
<td>preset-ibcf-ext-out-opt</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Type: Blacklist Actions: No options (passes on all)</td>
</tr>
<tr>
<td></td>
<td>preset-ibcf-int-out-mth</td>
<td>Type: Blacklist Actions: No methods rejected</td>
<td>preset-ibcf-int-out-opt</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Type: Blacklist Actions: No options (passes on all)</td>
</tr>
</tbody>
</table>
Passthrough of From, To, and Contact Headers

For Dialog-creating and Out-of-dialog requests, Cisco Unified Border Element (SP Edition) allows the From, To, and Contact header URIs to pass through without modifying them. For dialog headers, Cisco Unified Border Element (SP Edition) uses the values corresponding to those on the Out-of-dialog requests.

Passthrough of Request URI on REGISTER

Cisco Unified Border Element (SP Edition) allows the Request URI on a REGISTER message to pass through without modifying it.

Interworking with P-CSCF, I-CSCF, and S-CSCF

When performing the role of an IBCF gateway, Cisco Unified Border Element (SP Edition) allows the CSCF-specific headers on SIP messages to pass through.

Handling Messages from Untrusted Domains

When Cisco Unified Border Element (SP Edition) is acting as an IBCF entry point, it handles out-of-dialog requests from untrusted domains as follows:

- Cisco Unified Border Element (SP Edition) rejects all REGISTER requests with a 403 response.
- Cisco Unified Border Element (SP Edition) removes all P-Asserted-Identity headers, P-Access-Network-Info headers, P-Charging-Vector headers, and P-Charging-Function-Address headers from other requests.
- Cisco Unified Border Element (SP Edition) rejects requests if the router contains the Orig parameter.

Implementing IBCF Support

Configuring the Domain Names to Use for IBCF Adjacencies

**SUMMARY STEPS**

1. configure terminal
2. sbc service-name
3. sbe
4. sip home network identifier domain-name
5. sip encryption key string
6. adjacency sip adjacency-name
7. inherit profile preset-ibcf-internal
8. home network identifier domain-name
### Implementing IBCF Support

9. **encryption key** *string*

10. **exit**

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enables global configuration mode.</td>
</tr>
</tbody>
</table>

**Example:**

Router# configure terminal

| **Step 2** abc service-name | Enters the mode of an SBC service.  
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Use the <em>service-name</em> argument to define the name of the service.</td>
</tr>
<tr>
<td>abc mysbc</td>
<td></td>
</tr>
</tbody>
</table>

| **Step 3** sbe | Enters the mode of a SBE entity within an SBC service. |

**Example:**

Router(config)# sbe

| **Step 4** sip home network identifier domain-name | Configures the specified domain name as the global home network identifier for use in all SIP IBCF adjacencies.  
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Use the <em>domain-name</em> argument to specify the domain name of the SIP adjacency.</td>
</tr>
<tr>
<td>Router(config-sbc-sbe)# sip home network identifier mydomain.com</td>
<td></td>
</tr>
</tbody>
</table>

| **Step 5** sip encryption key string | Configures a global encryption key for all SIP IBCF adjacencies.  
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Use the <em>string</em> value to specify the encryption key to use for all SIP IBCF adjacencies.</td>
</tr>
<tr>
<td>Router(config-sbc-sbe)# encryption key code1</td>
<td></td>
</tr>
</tbody>
</table>

| **Step 6** adjacency sip adjacency-name | Enters the mode of an SBE SIP adjacency.  
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Use the <em>adjacency-name</em> argument to define the name of the SIP adjacency.</td>
</tr>
<tr>
<td>Router(config-sbc-sbe-adj-sip)# adjacency sip sipadj</td>
<td></td>
</tr>
</tbody>
</table>

| **Step 7** inherit profile preset-ibcf-internal | Configures a global inherit profile and specifies a preset IBCF internal profile. |

**Example:**

Router(config-sbe-adj-sip)# inherit profile preset-ibcf-internal

| **Step 8** home network identifier network-name | Configures a home network identifier on an IBCF adjacency.  
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Use the <em>network-name</em> argument to specify the name of the home network identifier.</td>
</tr>
<tr>
<td>Router(config-sbc-sbe-adj-sip)# home network identifier Cisco.com</td>
<td></td>
</tr>
</tbody>
</table>

---
### Implementing IBCF Support

#### Step 9
**encryption key** *string*

**Example:**
```
Router(config-sbc-sbe-adj-sip)# encryption key code2
```

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| encryption key *string* | Configures an encryption key on the SIP IBCF adjacency.  
  - Use the *string* argument to specify the encryption key for the SIP IBCF adjacency. |

#### Step 10
**exit**

**Example:**
```
Router(config-sbc-sbe-adj-sip)# exit
```

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>exit</td>
<td>Exits the SIP adjacency mode to the SBE mode.</td>
</tr>
</tbody>
</table>
IMS Rx, Diameter, and IMS Rf

The Cisco Unified Border Element (SP Edition) supports IP Multimedia Subsystem (IMS) Rx interfaces, Diameter protocol, and IMS Rf interfaces.

An IMS Rx is a Third Generation Partnership Project (3GPP) interface that runs between an application function and a Policy Charging and Rules Function (PCRF) in a 3GPP architecture.

The Diameter is an Authentication Authorization Accounting (AAA) protocol and is an enhanced version of the RADIUS (Remote Authentication Dial-In User Service) protocol.

An IMS Rf is an interface that runs between Charging Trigger Function (CTF) and Charging Data Function (CDF) in a 3GPP architecture.

Feature History for IMS Rx, Diameter, and IMS Rf

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 3.1S</td>
<td>• The IMS Rx Interfaces feature was introduced.</td>
</tr>
<tr>
<td></td>
<td>• The Diameter feature was introduced.</td>
</tr>
<tr>
<td>Cisco IOS XE Release 3.7S</td>
<td>The IMS Rf Billing Interface feature was introduced.</td>
</tr>
</tbody>
</table>

Contents

- Information About IMS Rx Interfaces, page 57-2
- Configuring IMS Rx, page 57-3
- Configuration Examples for IMS Rx, page 57-7
- Information About the Diameter Protocol in the SBC, page 57-8
- Configuring SBC Diameter Routing, page 57-9
- Configuration Examples for Diameter Routing, page 57-14
- Information About IMS Rf Billing Interfaces, page 57-16
- Configuring an IMS Rf Billing Interface, page 57-17
- Configuration Example for IMS Rf Billing Interface, page 57-20
Information About IMS Rx Interfaces

An IMS Rx interface is a 3GPP interface that runs between an application function and a Policy Charging and Rules Function (PCRF) in a 3GPP architecture. In this case, SBC is the application function.

SBC uses the Rx interface to communicate with the PCRF during call initiation and renegotiation to ensure that a call conforms to policy. SBC uses the Rx interface during registration to learn access network information.

The PCRF performs the following functions for SBC via an IMS Rx interface:

- Confirms that call media requests conform to the appropriate policy.
- Opens gates or pinholes in the media route, and specifies the appropriate QoS.
- Requests per-flow charging information when needed.
- Informs SBC of media-plane events.

An IMS Rx interface can be configured as a pure Rx environment or as a mixed Rx and media resource environment in unified SBC.

Features Supported

SBC can be deployed as the application function connecting to a PCRF via an Rx interface, in a mobile network, under an IMS or non-IMS environment. SBC supports the following requirements for these environments:

- Support for precondition call flows with Rx
- Support for late-INVITE and PRACK with Rx
- SIP late and early interworking in combination with Rx
- SIP PRACK and non-PRACK interworking in combination with Rx
- Support for session binding on registration
- SBC does not add any IMS-specific SIP headers to requests or responses in non-IMS environment, and does not add P-Charging-Vector or P-Access-Network-Info information
- SBC can also use an Rx interface to query a policy server to perform admission control for requests from subscribers on an access network in non-IMS environments.

Restrictions

- SBC does not provide preferred or alternate routes for SIP or DNS interfaces.
- SBC does not support use of Rx in combination with local call transfers.
- Lawful Intercept of media for calls using Rx is not possible.
- SBC does not support Packet Cable billing on Rx interfaces.

Call Failures

If the PCRF fails to respond to a request from SBC, SBC treats only the individual request as failed. Only fully established calls are maintained during redundant switchovers. Calls in the process of being set up are dropped.

Configuration

- See the "Configuring IMS Rx? section on page 57-3 for the procedure for configuring an IMS Rx Interface."
See the "Configuration Examples for Diameter Routing? section on page 57-14 for configuration examples of IMS Rx.

## Configuring IMS Rx

This section describes the following procedures:

- Configuring an IMS Rx Interface, page 57-3
- Configuring Media Service for IMS Rx, page 57-4
- Disabling Preliminary AAR Messages, page 57-6

## Configuring an IMS Rx Interface

Use the following procedure to configure an IMS Rx interface.

### SUMMARY STEPS

1. configure terminal
2. sbc sbc-name
3. sbe
4. adjacency sip adjacency-name
5. ims realm realm-name
6. ims rx
7. ims pani
8. end

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
</tbody>
</table>

**Example:**

Router# configure terminal

| **Step 2** sbc sbc-name | Creates the SBC service on Cisco Unified Border Element (SP Edition) and enters into SBC configuration mode. |

**Example:**

Router(config)# sbc MySBC

| **Step 3** sbe | Enters the mode of the signaling border element (SBE) function of the SBC. |

**Example:**

Router(config-sbc)# sbe
Configuring Media Service for IMS Rx

Use the following procedure to configure media service for IMS Rx.

**SUMMARY STEPS**

1. configure terminal
2. sbc sbc-name
3. sbe
4. cac-policy-set policy-set-id
5. cac-table table-name
6. table-type policy-set
7. entry entry-id
8. ims media-service
9. end
### Detailed Steps

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>2</td>
<td><code>sbc sbc-name</code></td>
<td>Creates the SBC service on Cisco Unified Border Element (SP Edition) and enters into SBC configuration mode.</td>
</tr>
<tr>
<td>3</td>
<td><code>sbe</code></td>
<td>Enters the mode of the signaling border element (SBE) function of the SBC.</td>
</tr>
<tr>
<td>4</td>
<td><code>cac-policy-set policy-set-id</code></td>
<td>Enters the mode of CAC policy set configuration within an SBE entity, creating a new policy set if necessary.</td>
</tr>
<tr>
<td>5</td>
<td><code>cac-table table-name</code></td>
<td>Enters the mode for configuration of an admission control table (creating one if necessary) within the context of an SBE policy set.</td>
</tr>
<tr>
<td>6</td>
<td><code>table-type policy-set</code></td>
<td>Configures a CAC table to allow the use of media resources and third party transcoding resources as well as Rx resources the table type of a CAC table within the context of an SBE policy set.</td>
</tr>
<tr>
<td>7</td>
<td><code>entry entry-id</code></td>
<td>Enters the mode to modify an entry in an admission control table.</td>
</tr>
<tr>
<td>8</td>
<td><code>ims media-service</code></td>
<td>(Optional) Configures a CAC table to allow the use of media resources and third party transcoding resources as well as Rx resources.</td>
</tr>
<tr>
<td>9</td>
<td><code>end</code></td>
<td>Exits configuration mode and returns to privileged EXEC mode.</td>
</tr>
</tbody>
</table>

Example:
- `Router# configure terminal`
- `Router(config)# sbc SBC1`
- `Router(config-sbc)# sbe`
- `Router(config-sbc-sbe)# cac-policy-set 1`
- `Router(config-sbc-sbe-cacpolicy)# cac-table testSecure`
- `Router(config-sbc-sbe-cacpolicy-cactable)# table-type policy-set`
- `Router(config-sbc-sbe-cacpolicy-cactable-entry)# entry 1`
- `Router(config-sbc-sbe-cacpolicy-cactable-entry)# ims media-service`
- `Router(config-sbc-sbe-enum-entry)# end`
Disabling Preliminary AAR Messages

Use the following procedure optionally to prevent preliminary AAR messages from being sent during an IMS Rx session.

SUMMARY STEPS

1. configure terminal
2. sbc sbc-name
3. sbe
4. cac-policy-set policy-set-id
5. cac-table table-name
6. table-type policy-set
7. entry entry-id
8. ims rx preliminary-aar-forbid
9. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc sbc-name</td>
<td>Creates the SBC service on Cisco Unified Border Element (SP Edition) and enters into SBC configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# sbc SBC1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Enters the mode of the signaling border element (SBE) function of the SBC.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> cac-policy-set policy-set-id</td>
<td>Enters the mode of CAC policy set configuration within an SBE entity, creating a new policy set if necessary.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe)# cac-policy-set 1</td>
<td>policy-set-id—Integer chosen by the user to identify the policy set. The range is 1 to 2147483647.</td>
</tr>
<tr>
<td><strong>Step 5</strong> cac-table table-name</td>
<td>Enters the mode for configuration of an admission control table (creating one if necessary) within the context of an SBE policy set.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-cacpolicy)# cac-table testSecure</td>
<td>table-name—Name of the admission control table.</td>
</tr>
</tbody>
</table>
### Configuration Examples for IMS Rx

This section provides the following examples:

- Configuration Example for IMS Rx Interface, page 57-7
- Configuration Example for IMS Rx Media Service, page 57-7
- Configuration Example for Disabling Preliminary AAR Messages, page 57-8

#### Configuration Example for IMS Rx Interface

The following example shows how to configure an IMS Rx interface:

```bash
Router# configure terminal
Router(config)# sbc mySBC
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency sip A_1
Router(config-sbc-sbe-adj-sip)# ims realm Realm_1
Router(config-sbc-sbe-adj-sip)# ims rx
Router(config-sbc-sbe-adj-sip)# ims pani
Router(config-sbc-sbe-adj-sip)# end
```

#### Configuration Example for IMS Rx Media Service

The following example shows how to configure media service for IMS Rx:

```bash
Router# configure terminal
Router(config)# sbc MySBC
Router(config-sbc)# sbe
```

---

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 6</strong> table-type policy-set</td>
<td>Configures a CAC table to allow the use of media resources and third party transcoding resources as well as Rx resources the table type of a CAC table within the context of an SBE policy set.</td>
</tr>
<tr>
<td>Example: Router(config-sbc-sbe-cacpolicy-cactable)# table-type policy-set</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> entry entry-id</td>
<td>Enters the mode to modify an entry in an admission control table.</td>
</tr>
<tr>
<td>Example: Router(config-sbc-sbe-cacpolicy-cactable)# entry 1</td>
<td>entry-id—Specifies the table entry.</td>
</tr>
<tr>
<td><strong>Step 8</strong> ims rx preliminary-aar-forbid</td>
<td>Prevents preliminary AAR messages from being sent during an IMS Rx session.</td>
</tr>
<tr>
<td>Example: Router(config-sbc-sbe-cacpolicy-cactable-entry)# ims rx preliminary-aar-forbid</td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong> end</td>
<td>Exits configuration mode and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td>Example: Router(config-sbc-sbe-enum-entry)# end</td>
<td></td>
</tr>
</tbody>
</table>
Information About the Diameter Protocol in the SBC

Diameter is an Authentication Authorization Accounting (AAA) protocol and is an enhanced version of the RADIUS (Remote Authentication Dial-In User Service) protocol. Diameter is the protocol of choice for the next generation IMS network developed by 3GPP.

When the Diameter protocol is implemented on a network, the Policy Charging and Rules Function (PCRF) acts as the Diameter server and the Application Function (AF), in our case SBC, acts as the Diameter client. SBC performs the functions of an IMS Rx Diameter client application and handles policy information and media reservations at the border of an access network.

SBC Diameter provides users with the option of configuring of either of two types of routing:

- Host-based routing
- Realm-based routing where multiple peers can be configured

Interfaces are referred as reference points in IMS. Reference points are named using unique acronyms, such as Rx (receiving reference point).

Features Supported

The following features are supported by SBC Diameter:

- SBC Diameter runs over TCP.
- SBC Diameter uses IPv4 addressing only.
- SBC Diameter supports multiple peers per realm.
- SBC Diameter supports redundancy switchover of Diameter peers as follows:
  - All Diameter messages are sent to the primary peer of the realm by default.
  - If the primary peer fails, Diameter switches to a secondary peer and retransmits all pending messages
Restrictions
SBC Diameter has the following restrictions:

- SBC Diameter does not replicate states or outstanding requests during redundancy switchovers. All states and outstanding requests are lost after a switchover from a failed active connection to a backup connection.
- SBC Diameter does not support IPv6 addressing.
- IPv6 is not supported.

Configuration
See the Configuring SBC Diameter Routing section on page 57-9 for the procedure for configuring the Diameter protocol in SBC.

See the Configuration Examples for Diameter Routing section on page 57-14 for configuration examples of the Diameter protocol in SBC.

Configuring SBC Diameter Routing

This section provides two routing configurations:

- Configuring Diameter Host-Based Routing, page 57-9
- Configuring Diameter Realm-Based Routing, page 57-11

Configuring Diameter Host-Based Routing

Use the following procedure to configure Diameter host-based routing. This procedure sets up an Rx adjacency first and then the Diameter host-based routing.

SUMMARY STEPS

1. configure terminal
2. sbc sbc-name
3. sbe
4. adjacency sip adjacency-name
5. ims realm realm-name
6. ims rx pcrf pcrf-name
7. ims pani [received | rx | received rx | rx received]
8. exit
9. diameter
10. origin-realm realm-name
11. origin-host host-name
12. activate
13. end
14. show sbc sbc-name sbe diameter
15. `show sbc sbc-name sbe diameter peers peer-name`

16. `show sbc sbc-name sbe diameter stats`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>configure terminal</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>Router# configure terminal</code></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><code>sbc sbc-name</code></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>Router(config)# sbc MySBC</code></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><code>sbe</code></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>Router(config-sbc)# sbe</code></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td><code>adjacency sip adjacency-name</code></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>Router(config-sbc-sbe) adjacency sip Adj_1</code></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td><code>ims realm realm-name</code></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>Router(config-sbc-sbe-adj-sip)# ims realm Rx_Realm_1</code></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td><code>ims rx pcrf pcrf-name</code></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>Router(config-sbc-sbe-adj-sip)# ims rx pcrf cisco.com</code></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>`ims pani [ received</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>Router(config-sbc-sbe-adj-sip)# ims pani rx received</code></td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td><code>exit</code></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>Router(config-sbc-sbe-enum)# exit</code></td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td><code>diameter</code></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>Router(config-sbc-sbe)# diameter</code></td>
</tr>
</tbody>
</table>
Configuring SBC Diameter Routing

Use the following procedure to configure Diameter realm-based routing.

**SUMMARY STEPS**

1. configure terminal
2. sbc sbc-name
3. sbe
4. adjacency sip adjacency-name
5. ims realm realm-name
6. ims rx
7. exit

---

**Step 10**

**Command or Action**: `origin-realm realm-name`

**Purpose**: Configures the name of SBC’s local realm for diameter messages.

**Example**: `Router(config-sbc-sbe-diameter)# origin-realm cisco.com`

**Step 11**

**Command or Action**: `origin-host host-name`

**Purpose**: Configures the name of SBC’s local host for diameter messages.

**Example**: `Router(config-sbc-sbe-diameter)# origin-host sbc.cisco.com`

**Step 12**

**Command or Action**: `activate`

**Purpose**: Activates Diameter host-based routing.

**Example**: `Router(config-sbc-sbe-enum)# activate`

**Step 13**

**Command or Action**: `end`

**Purpose**: Exits configuration mode and returns to privileged EXEC mode.

**Example**: `Router(config-sbc-sbe-enum-entry)# end`

**Step 14**

**Command or Action**: `show sbc sbc-name sbe diameter`

**Purpose**: Displays the local configuration information for Diameter.

**Example**: `Router# show sbc MySBC sbe diameter`

**Step 15**

**Command or Action**: `show sbc sbc-name sbe diameter peers peer-name`

**Purpose**: Displays the configuration information for IMS peers.

**Example**: `Router# show sbc MySBC sbe diameter peers Peer1`

**Step 16**

**Command or Action**: `show sbc sbc-name sbe diameter stats`

**Purpose**: Displays the transport statistics for an IMS peer.

**Example**: `Router# show sbc MySBC sbe diameter stats`
### Chapter 57  IMS Rx, Diameter, and IMS Rf

#### Configuring SBC Diameter Routing

8. `diameter`
9. `origin-realm realm-name`
10. `origin-host host-name`
11. `peer peer-name ipv4 ipv4-address`
12. `peer peer-name ipv4 ipv4-address`
13. `realm realm-name [app rx] peer peer-name [priority priority]`
14. `realm realm-name [app rx] peer peer-name [priority priority]`
15. `activate`
16. `end`
17. `show sbc sbc-name sbe diameter peers`
18. `show sbc sbc-name sbe diameter peers peer-name`
19. `show sbc sbc-name sbe diameter peers peer-name`

### Detailed Steps

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1**
  
  **configure terminal**
  
  **Example:**
  
  `Router# configure terminal`
  
  Enters global configuration mode. |
| **Step 2**
  
  **sbc sbc-name**
  
  **Example:**
  
  `Router(config)# sbc MySBC`
  
  Creates the SBC service on Cisco Unified Border Element (SP Edition) and enters into SBC configuration mode. |
| **Step 3**
  
  **sbe**
  
  **Example:**
  
  `Router(config-sbc)# sbe`
  
  Enters the mode of the signaling border element (SBE) function of the SBC. |
| **Step 4**
  
  **adjacency sip adjacency-name**
  
  **Example:**
  
  `Router(config-sbc-sbe)# adjacency sip Adj_1`
  
  Enters the mode of an SBE SIP adjacency. |
| **Step 5**
  
  **ims realm realm-name**
  
  **Example:**
  
  `Router(config-sbc-sbe-adj-sip)# ims realm Rx_Realm_1`
  
  Creates an IMS realm for the Rx. |
| **Step 6**
  
  **ims rx**
  
  **Example:**
  
  `Router(config-sbc-sbe-adj-sip)# ims rx pcrf cisco.com`
  
  Configures an IMS Rx reference point on this SIP adjacency. |
<table>
<thead>
<tr>
<th>Step 7</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>exit</strong></td>
<td>Exits to the previous mode.</td>
</tr>
</tbody>
</table>

**Example:**
```
Router(config-sbc-sbe-enum)# exit
```

<table>
<thead>
<tr>
<th>Step 8</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>diameter</strong></td>
<td>Enters the Diameter configuration mode.</td>
</tr>
</tbody>
</table>

**Example:**
```
Router(config-sbc-sbe)# diameter
```

<table>
<thead>
<tr>
<th>Step 9</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>origin-realm realm-name</strong></td>
<td>Configures the domain name of an IMS local realm.</td>
</tr>
</tbody>
</table>

**Example:**
```
Router(config-sbc-sbe-diameter)# origin-realm cisco.com
```

<table>
<thead>
<tr>
<th>Step 10</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>origin-host host-name</strong></td>
<td>Configures the domain name of the local IMS host.</td>
</tr>
</tbody>
</table>

**Example:**
```
Router(config-sbc-sbe-diameter)# origin-host sbc.cisco.com
```

<table>
<thead>
<tr>
<th>Step 11</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>peer peer-name ipv4 ipv4-address</strong></td>
<td>Configures the name and IPv4 address of peerA.</td>
</tr>
</tbody>
</table>

**Example:**
```
Router(config-sbc-sbe-diameter)# peer peerA address ipv4 1.2.3.4
```

<table>
<thead>
<tr>
<th>Step 12</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>peer peer-name ipv4 ipv4-address</strong></td>
<td>Configures the name and IPv4 address of peerB.</td>
</tr>
</tbody>
</table>

**Example:**
```
Router(config-sbc-sbe-diameter)# peer peerB address ipv4 1.2.3.5
```

<table>
<thead>
<tr>
<th>Step 13</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>realm realm-name</strong> (app rx) peer peer-name [priority priority]</td>
<td>Configures a peer and assign the peer to the realm.</td>
</tr>
</tbody>
</table>

**Example:**
```
Router(config-sbc-sbe-diameter)# realm test.com app rx peer peerA
```

<table>
<thead>
<tr>
<th>Step 14</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>realm realm-name</strong> (app rx) peer peer-name [priority priority]</td>
<td>Configures another peer and assign the peer to the realm.</td>
</tr>
</tbody>
</table>

**Example:**
```
Router(config-sbc-sbe-diameter)# realm test.com app rx peer peerB priority 10
```

<table>
<thead>
<tr>
<th>Step 15</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>activate</strong></td>
<td>Activates Diameter realm-based routing.</td>
</tr>
</tbody>
</table>

**Example:**
```
Router(config-sbc-sbe-enum)# activate
```
Step 16

```
end
```

**Example:**
```
Router(config-sbc-sbe-enum-entry)# end
```

**Purpose:** Exits configuration mode and returns to privileged EXEC mode.

Step 17

```
show sbc sbc-name sbe diameter peers
```

**Example:**
```
Router# show sbc MySBC sbe diameter peers
```

**Purpose:** Displays the configuration information for all IMS peers.

Step 18

```
show sbc sbc-name sbe diameter peers peer-name
```

**Example:**
```
Router# show sbc MySBC sbe diameter peers peerA
```

**Purpose:** Displays the configuration information for peerA.

Step 19

```
show sbc sbc-name sbe diameter peers peer-name
```

**Example:**
```
Router# show sbc MySBC sbe diameter peers peerB
```

**Purpose:** Displays the configuration information for peerB.

---

**Configuration Examples for Diameter Routing**

This section provides the following examples:

- Configuration Example for Diameter Host-Based Routing, page 57-14
- Configuration Example for Diameter Realm-Based Routing, page 57-15

---

**Configuration Example for Diameter Host-Based Routing**

The following example shows how to configure Diameter host-based routing:
```
Router# configure terminal
Router(config)# sbc MySBC
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency sip Adj_1
Router(config-sbc-sbe-adj-sip)# ims realm Rx_Realm_1
Router(config-sbc-sbe-adj-sip)# ims rx pcf cisco.com
Router(config-sbc-sbe-adj-sip)# ims pani
Router(config-sbc-sbe-enum)# exit
Router(config-sbc-sbe)# diameter
Router(config-sbc-sbe-diameter)# origin-realm cisco.com
Router(config-sbc-sbe-diameter)# origin-host sbc.cisco.com
Router(config-sbc-sbe-enum)# activate
Router(config-sbc-sbe-enum-entry)# end
Router# show sbc MySBC sbe diameter
Router# show sbc MySBC sbe diameter peers Peer1
Router# show sbc MySBC sbe diameter stats
```
Configuration Example for Diameter Realm-Based Routing

The following example shows how to configure Diameter realm-based routing:

```
Router# configure terminal
Router(config)# sbc MySBC
Router(config-sbc)# sbe
Router(config-sbc-sbe)# adjacency sip Adj_1
Router(config-sbc-sbe-adj-sip)# ims realm Rx_Realm_1
Router(config-sbc-sbe-adj-sip)# ims rx
Router(config-sbc-sbe-enum)# exit
Router(config-sbc-sbe)# diameter
Router(config-sbc-sbe-diameter)# origin-realm cisco.com
Router(config-sbc-sbe-diameter)# origin-host sbc.cisco.com
Router(config-sbc-sbe-diameter)# peer peerA address ipv4 1.2.3.4
Router(config-sbc-sbe-diameter)# peer peerB address ipv4 1.2.3.5
Router(config-sbc-sbe-diameter)# realm test.com app rx peer peerA
Router(config-sbc-sbe-diameter)# realm test.com app rx peer peerB priority 10
Router(config-sbc-sbe-enum)# activate
Router(config-sbc-sbe-enum-entry)# end
Router# show sbc MySBC sbe diameter peers
Router# show sbc MySBC sbe diameter peers peerA
Router# show sbc MySBC sbe diameter peers peerB
```

**Note**

You can use the following, existing ASR1000 IPSEC functionality to provide secure Diameter protocol transport:

```
crypto isakmp policy 1
  encr aes
  authentication pre-share
  group 2

crypto isakmp key cisco123 address 0.0.0.0 0.0.0.0

crypto ipsec transform-set testcpoc esp-des esp-md5-hmac

crypto map diamap 10 ipsec-isakmp
  set peer 192.68.9.1
  set security-association lifetime kilobytes 536870912
  set transform-set testcpoc
  match address 199

access-list 199 permit ip 192.169.0.0 0.0.255.255 193.169.0.0 0.0.255.255

interface SBC01
  ip address 192.68.9.2 255.255.255.0

crypto map diamap
```
Information About IMS Rf Billing Interfaces

The SBC supports Rf billing interfaces for SIP-to-SIP calls when operating as a Proxy Call Session Control Function (P-CSCF) and as an Interconnection Border Control Function (IBCF). The Charging Trigger Function (CTF) in the SBC uses an Rf billing interface to provide offline charging information to the billing domain in an IMS network. The Rf billing interface uses the Diameter protocol for sending billing information to the Charging Data Function (CDF). Offline charging is used for network services that are paid periodically, for example, a user may have a subscription for voice calls that is paid for on a monthly basis.

In IMS, billing information originates from the CTF. The CTF sends Accounting Request (ACR) messages containing billing information to the CDF, which collates this information into event-based and session-based Call Detail Record (CDR) files. The CDF then passes the files to the Charging Gateway Function (CGF), which is responsible for nonvolatile storage of the CDRs and for other functions such as, duplicate detection, error correction, and filtering. The CGF transfers the files to the billing domain for eventual account reconciliation. This final transfer is not time sensitive and can occur in batch mode. The billing domain uses the CDR to charge for the services used.

Offline Charging Events

For both event-based charging and session-based charging, the CTF supports the accounting state machine. The task of reporting offline charging events to the CDF is managed through a Diameter Accounting Request (ACR) message. The IMS Rf interface supports the ACR event types described in Table 57-1.

<table>
<thead>
<tr>
<th>Event Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>START</td>
<td>Starts an accounting session.</td>
</tr>
<tr>
<td>INTERIM</td>
<td>Updates an accounting session.</td>
</tr>
<tr>
<td>STOP</td>
<td>Stops an accounting session.</td>
</tr>
<tr>
<td>EVENT</td>
<td>Indicates a one-time accounting event.</td>
</tr>
</tbody>
</table>

The START, INTERIM, and STOP event types are used for session-based charging. The EVENT type is used either for event-based charging or to indicate a failed attempt at establishing a session.

Rf Billing Error Handling

This section describes how the SBC handles the various types of Rf billing errors.

CDF Connection Failure

If the connection to the primary CDF is broken, the SBC sends the corresponding charging information to the secondary CDF (if present). If statically configured CDFs are used, the secondary CDF is the redundant peer of the next highest priority. If the dynamic CDF discovery task is performed, the secondary CDF is the address in the next ccf parameter in the P-Charging-Function-Address header. This process continues until a CDF responds, or there are no more CDFs. In the latter scenario, if an appropriate file system is available, the charging messages are stored in the nonvolatile memory until the CDF connection is restored. The connection to any of the available CDFs has no impact on the call setup.
No Reply from CDF
Because DIAMETER messages are transmitted over TCP or Stream Control Transmission Protocol (SCTP), a missing Accounting Answer response to an ACR must indicate that a connection is going down. In such a scenario, the procedure described in CDF Connection Failure section is followed.

Failure Response from CDF
The CDF can return any failure encountered while collecting billing information from the SBC, in the ACA message, even though the connection to the peer is active.
If the failure return code is DIAMETER_UNABLE_TO_DELIVER, this message is cached in nonvolatile memory and follows the procedure described in CDF Connection Failure section.
If the failure return code is any other value, a PD log is created to convey this information to the user, but no other action is taken.

Duplicate Detection
The SBC does not retransmit DIAMETER requests because the underlying TCP transport handles such requests. The CDF does not handle duplicate requests from the SBC.

CDF Detected Failure
If the SBC fails over, some Rf sessions may not be closed correctly, for example, when a call is set up during failover. The CDF must close CDRs pertaining to a particular session if it detects that ACRs are not received within a certain period.

Restrictions for IMS Rf Billing Interfaces
The IMS Rf Billing Interfaces feature has the following restrictions:
- The SBC does not support Rf billing for SIP-to-H.323 calls and H.323-to-H.323 calls.
- The SBC does not support Rf billing in a non-IMS network.
- The SBC does not supply the PS-Information attribute-value pairs (AVP) on its messages. Therefore, the SBC does not send the Cisco Gateway GPRS Support Node (GGSN)-Address AVP.
- The SBC does not supply the Third Generation Partnership Project (3GPP)-Charging-ID AVP.

Configuring an IMS Rf Billing Interface
Use the following procedure to configure an IMS Rf billing interface.

SUMMARY STEPS
1. configure terminal
2. sbc sbc-name
3. sbe
4. adjacency sip adjacency-name
5. ims rf
6. ims realm realm-name
7. exit
### Chapter 57      IMS Rx, Diameter, and IMS Rf

#### Configuring an IMS Rf Billing Interface

8. **billing**  
9. **method 3gpp-rf**  
10. **rf index**  
11. **origin-host host-name**  
12. **origin.realm realm-name**  
13. **realm realm-name [usePCFAHeader | cdf cdf-name [FQDN FQDN-name | ipv4 ipv4-address | vpn vpn-name] [port port-number] [priority priority-number]]**  
14. **attach**  
15. **activate**  
16. **end**  
17. **show sbc sbc-name sbe adjacencies adjacency-name [authentication-realms | detail | peers]**  
18. **show sbc sbc-name sbe billing instance [instance-index] [rf {realms [realm-name current5mins]} | {cdfs [cdf-name]}]**

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters the global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc sbc-name</td>
<td>Creates the SBC service on the Cisco Unified Border Element (SP Edition) and enters the SBC configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# sbc MySBC</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Enters the mode of the signaling border element (SBE) function of the SBC.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> adjacency sip adjacency-name</td>
<td>Enters the SBE SIP adjacency mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe) adjacency sip A_1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> ims rf</td>
<td>Configures an IMS Rf interface for access adjacency.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-adj-sip)# ims rf</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> ims realm realm-name</td>
<td>Configures an IMS realm for use by an IMS Rf interface.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe-adj-sip)# ims realm Realm_1</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action | Purpose
--- | ---
**Step 7** | exit
**Example:**
Router(config-sbc-sbe-adj-sip)# exit
| Exits the SBE SIP adjacency mode.

**Step 8** | billing
**Example:**
Router(config-sbc-sbe)# billing
| Configures the IMS Rf billing method.

**Step 9** | method 3gpp-rf
**Example:**
Router(config-sbc-sbe-billing)# method 3gpp-rf
| Enables the 3GPP Rf billing method on the SBC.

**Step 10** | rf index
**Example:**
Router(config-sbc-sbe-billing)# rf 0
| Creates a new Rf billing instance.

**Step 11** | origin-host host-name
**Example:**
Router(config-sbc-sbe-billing-rf)# origin-host sbc.com
| Configures the domain name of an IMS local host. This value is displayed in the diameter Origin-Host AVP.

**Step 12** | origin-realm realm-name
**Example:**
Router(config-sbc-sbe-billing-rf)# origin-realm cisco.com
| Configures the domain name of an IMS local realm. This value is displayed in the diameter Origin-Realm AVP.

**Step 13** | realm realm-name [usePCFAMheader | cdf cdf-name (FQDN FQDN-name | ipv4 ipv4-address | vpn vpn-name) [port port-number] [priority priority-number]]
**Example:**
Router(config-sbc-sbe-billing-rf)# realm cisco.com cdf cdf1 ipv4 192.0.2.1 port 3688
| Enables dynamic CDF detection.

**Step 14** | attach
**Example:**
Router(config-sbc-sbe-billing-rf)# attach
| Attaches an adjacency to an account on the SBE.

**Step 15** | activate
**Example:**
Router(config-sbc-sbe-billing-rf)# activate
| Activates billing after it is configured.

**Step 16** | end
**Example:**
Router(config-sbc-sbe-billing-rf)# end
| Exits the configuration mode and returns to the privileged EXEC mode.
The following example shows how to configure the IMS Rf Billing Interface feature:

```
configure terminal
sbc MySBC
sbe
adjacency sip test
ims rf
ims realm cisco.com
billing
method 3GPP-RF
rf 0
  orig-host sbc.com
  orig-realm cisco.com
rf 0 realm cisco.com cdf cdf1 ipv4 1.2.3.4 port 3688
rf 0 realm cisco.com cdf cdf2 cdf.cisco.com priority 2
attach
activate
end
```

The following is a sample output of the `show sbc sbe billing instance` command:

```
Router# show sbc asr sbe billing instance 6 rf realms realm1 current5mins

Billing Manager Information:
Local IP address: 3.3.3.3
LDR check time: 0:0
Method rf
Admin Status: UP
Operation Status: UP

Billing Methods
Instance: 1
Type: 3GPP-RF
Transport Mechanism Status: FAILED
Active Calls Billed: 0
Deact-mode: abort
Admin Status: UP
Operation Status: UP
LDR check time: 24:0
Origin Host: yfasr.open-ims.test
```
Origin Realm: open-ims.test
CALEA IRI Interface Support

The Communications Assistance for Law Enforcement Act (CALEA) intercept-related information (IRI) Interface Support feature enables service providers to define a legal warrant on VoIP endpoints to gather both signaling and media content information. The CALEA IRI Interface Support feature is based on PacketCable 1.5 standard specifications.

The CALEA IRI Interface Support feature is applicable to both Session Initiation Protocol (SIP) and H.323 calls in a unified Session Border Controller (SBC) configuration. It is not, however, applicable to distributed SBC.

Cisco Unified Border Element (SP Edition) was formerly known as Integrated Session Border Controller and may be commonly referred to in this document as the SBC.

For a complete description of the commands used in this chapter, refer to the Cisco Unified Border Element (SP Edition) Command Reference: Unified Model at:


For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

Feature History for CALEA IRI Interface Support

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 3.1S</td>
<td>The CALEA IRI Interface Support feature was introduced on Cisco ASR 1000 Series Routers.</td>
</tr>
</tbody>
</table>

Contents

This module contains the following sections:

- Information About CALEA IRI Interface Support, page 58-2
- Restrictions for Implementing CALEA IRI Interface Support, page 58-12
- Implementing CALEA IRI Interface Support, page 58-13
The SBC can be used for the dual functions of Intercepting Control Element (ICE) and Intercepting Network Element (INE). You can place a request for a warrant using the Simple Network Management Protocol (SNMP) interface. The Cisco ASR 1000 series router responds with PacketCable1.5 messages and with replicated IP/UDP/RTP media packets, as required by the warrant.

You can also define the endpoint match using username, phone number, or SIP-Uniform Resource Identifier (URI). In addition, you can set up pen, trace, pen-and-trace, or intercept type of warrant.

You can define the VoIP endpoint information along with mediation device information using Simple Network Management Protocol Version 3 (SNMPv3) MIBs. The VoIP signaling information is sent from a router to a mediation device. In addition, the media content is tapped, replicated, encapsulated, and sent to the mediation device in real time.

Define the warrant by providing only the VoIP endpoint information. A Cisco ASR 1000 Series Router determines the local pinhole being used for a particular call, and replicates the call content to the mediation device. In addition, you can define the warrant by requesting only the call signaling-related information using PacketCable1.5 Event messages (IRI).

In the context of calls coming in on an adjacency, with the inherit profile set to preset-access, the source information from the SIP header will be used to match the configured warrants. In the context of the calls coming in on an adjacency, with the inherit profile set to preset-core, the destination information from the SIP header will be used to match the configured warrants. However, the provider can override these rules by configuring the `warrant match-order` command on the adjacencies.

For a registered SIP endpoint, we recommend setting `cvoiptapStreamMatchType` to URI.

When the VoIP call gets tapped, the Cisco ASR 1000 series router sends the locally generated unique Call Content Connection ID (CCCID) information using the RADIUS message. The same CCCID information is then used to encapsulate the media IP packet. An mediation device can use the CCCID information to correlate the signaling and media information. The VoIP LI warrant information can be retrieved using a secure SNMPv3 interface.

For each INTERCEPT, a unique IRI stream with CCCID information is present.

In a network setup of multiple Cisco ASR 1000 series routers, the CALEA IRI Interface Support feature is designed to tap the information on the router that is closest to the endpoint under surveillance.

This section contains the following information pertaining to the CALEA IRI Interface Support feature:

- CALEA IRI Interface Support Flow, page 58-3
- SNMP Row Indices, page 58-4
- Tap Interfaces, page 58-4
CALEA IRI Interface Support Flow

Figure 58-1 shows the flow of the CALEA IRI Interface Support feature.

**Figure 58-1  Flow of the CALEA IRI Interface Support Feature**

The steps pertaining to the flow of the CALEA IRI Interface Support feature are as follows:

1. Provisioning of mediation device information and VoIP warrant is done as a combination of SNMPv3 and IOS CLI commands on the Cisco ASR 1000 series router.
2. The calling party originates the call.
3. If a warrant matches the signaling parameters, RADIUS messages are sent to the mediation device. The message contains the unique CCCID generated by the Cisco ASR 1000 series router.
4. The party that was called answers, and the media information starts flowing through the Cisco ASR 1000 series router.
5. The Cisco ASR 1000 series router replicates the media information, and sends it to the mediation device.
SNMP Row Indices

Figure 58-2 represents the SNMP table and rows. There are two independent mediation device rows. The GenericStream and VoIP TAP MIB rows are the children of the IRI MD row. There is a CCIndex field in the VoIP TAP MIB row that captures the relationship with CC MD MIB row. A one-to-one relationship also exists between GenericStream and VoIP TAP MIB rows.

![Figure 58-2 SNMP Row Indices]

Tap Interfaces

This section describes the following Tap interfaces:

- **IRI Interface**, page 58-5
- **CC Interface**, page 58-11
IRI Interface

The PacketCable 1.5 standard specifications for Electronic Surveillance contains the packet definition for all IRI-related messages. Table 58-1 details the supported call event messages that are sent for each Tapped Call.

Table 58-1  Supported Call Event Messages

<table>
<thead>
<tr>
<th>Event Message</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signaling_Start</td>
<td>Sent when signalling has commenced (inbound), and when it is about to commence (outbound), for example, received INVITE on inbound, and about to send INVITE on outbound for a SIP endpoint.</td>
</tr>
<tr>
<td>QoS_Reserve</td>
<td>Sent for the inbound leg when the inbound QoS is reserved, and for the outbound leg when the outbound QoS is reserved.</td>
</tr>
<tr>
<td>Call_Answer</td>
<td>Indicates that the terminating party has answered, and that media has started. This message is sent for both the legs simultaneously.</td>
</tr>
<tr>
<td>QoS_Commit</td>
<td>Sent when QoS is committed by the SBC. This message is sent for both the legs at the same time.</td>
</tr>
<tr>
<td>Call_Disconnect</td>
<td>Sent when a call has been terminated, and media has ceased flowing. The message is sent for both the legs at the same time.</td>
</tr>
<tr>
<td>QoS_Release</td>
<td>Sent when the QoS is released by the SBC. Sent for both the legs at the same time.</td>
</tr>
<tr>
<td>Signaling_Stop</td>
<td>Sent when signaling is complete for each party in the call. The event is generated once for each party after the last signaling message is sent.</td>
</tr>
<tr>
<td>Media_Report</td>
<td>Sent by the SBC whenever a flow is created, modified, and released.</td>
</tr>
<tr>
<td>Surveillance_Stop</td>
<td>Sent by the SBC to indicate the end of the IRI or CC tapping or both. Generally, this means the end of a call.</td>
</tr>
<tr>
<td>Redirection</td>
<td>Sent by the SBC when a call has been transferred, either due to a 3XX redirect response, or a SIP REFER request.</td>
</tr>
</tbody>
</table>

Call Event Messages

Table 58-2 details the Signaling_Start message attributes that are supported and sent when the SBC has information that the destination is routable and the originating endpoint is allowed to make the call.

Table 58-2  Signaling_Start Message Attributes

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>EM_Header</td>
<td>Common header attribute.</td>
</tr>
<tr>
<td>Direction_Indicator</td>
<td>Specifies if the device represents an originating or terminating part of a call. 1—originating 2—terminating</td>
</tr>
</tbody>
</table>
Table 58-2  Signaling_Start Message Attributes (continued)

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTA_Endpoint_Name</td>
<td>The SBC has no direct contact with the MTA. By default, the value is set to MTA Endpoint. Alternatively, the attribute could be configured to report adjacency or signalling address information.</td>
</tr>
<tr>
<td>Calling_Party_Number</td>
<td>The number of the calling party (if available). In the SBC, this is the canonical format of the number after inbound number translations, if any, and before the routing.</td>
</tr>
<tr>
<td>Called_Party_Number</td>
<td>The number of the called party (always present). In the SBC, this is the canonical format of the number after any inbound number translation and before routing.</td>
</tr>
<tr>
<td>Routing_Number</td>
<td>Indicates a routable number (always present).</td>
</tr>
<tr>
<td>User_Input</td>
<td>The number of the called party prior to any translation performed during inbound number analysis.</td>
</tr>
<tr>
<td>Translation_Input</td>
<td>The number of the called party after inbound number analysis and before routing, if different from the value supplied in User_Input.</td>
</tr>
<tr>
<td>Redirected_From_Info</td>
<td>If originating an INVITE in response to a 3XX or a REFER, the attribute is set to the previous destination of the call (the sender of the 3XX or REFER), the initial destination of the call (if there are multiple redirections), and the number of redirections so far on the call.</td>
</tr>
<tr>
<td>Carrier_Identification_Code</td>
<td>The Carrier Identification Code associated with this call.</td>
</tr>
<tr>
<td>Trunk_Group_ID</td>
<td>Trunk_Type set to 9. Signaling type is not specified. Trunk_Group_ID set to the Trunk Group ID associated with the side of the call that is being tapped.</td>
</tr>
</tbody>
</table>

The following Signaling_Start message attributes are not included in the message:

- Attribute Name
- Location_Routing_Number
- Intl_Code
- Dial_Around_Code
- Jurisdiction_Information_Parameter
- Ported_In_Calling_Number
- Ported_In_Called_Number
- Called_Party_NP_source
- Calling_Party_NP_source
- Billing_Type
- Electronic_Surveillance_Indication

Table 58-3 details the QoS_Reserve message attributes. This message is generated when the SBC has reserved bandwidth (QoS) on the network. If the reserved bandwidth changes, QoS_Reserve and QoS_Commit messages are generated anew.

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>EM_Header</td>
<td>Common header attribute.</td>
</tr>
<tr>
<td>QoS_Descriptor</td>
<td>Similar to the description of the QoS_Reserve message.</td>
</tr>
<tr>
<td>MTA_UDP_Portnum</td>
<td>The UDP port number on the network element endpoint.</td>
</tr>
<tr>
<td></td>
<td>Because the SBC has no direct contact with the MTA, the attribute is set to 0.</td>
</tr>
<tr>
<td>Flow_Direction</td>
<td>1—upstream</td>
</tr>
<tr>
<td></td>
<td>2—downstream</td>
</tr>
<tr>
<td>SF_ID</td>
<td>A Data-over-Cable Service Interface Specifications-specific attribute that is required, and generated by the CMTS in a PacketCable architecture. Because the SBC does not support DOCSIS, this attribute is always 0.</td>
</tr>
<tr>
<td>CCC_ID</td>
<td>The local CCC ID for this call. It is included if CC tapping is being done on the call.</td>
</tr>
</tbody>
</table>

Table 58-4 details the Call_Answer message attributes. This message indicates the earliest point at which two-way media is established. The SBC sends the message to the billing servers when the SBC is notified that the called party has answered the call.

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>EM_Header</td>
<td>Common header attribute.</td>
</tr>
<tr>
<td>Charge_Number</td>
<td>The charge number during collect call, calling-card call, call billed to a third party, and so on. For the SBC, this is the calling number, unless the call has been diverted. The diverted call has a Diverted-By number.</td>
</tr>
<tr>
<td>Related_Call_Billing_Correlation_ID</td>
<td>The billing correlation ID (BCID) assigned to the leg from the terminating network element. The SBC does not share the BCID and financial entity ID (FEID) information with other network elements.</td>
</tr>
</tbody>
</table>

**Note**
The FEID attribute is not sent in a Call_Answer message.
Table 58-5 details the QoS_Commit message attributes. This message is sent by the SBC when the gate bandwidth is committed. This message is sent after a QoS_Reserve message that has been sent previously.

**Table 58-5 QoS_Commit Message Attributes**

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>EM_Header</td>
<td>Common header attribute containing timestamp and BCID.</td>
</tr>
<tr>
<td>MTA_UDP_Portnum</td>
<td>The UDP port number on the network element endpoint. Because the SBC has no direct contact with the MTA, so the attribute is set to 0.</td>
</tr>
<tr>
<td>Flow_Direction</td>
<td>1—upstream</td>
</tr>
<tr>
<td></td>
<td>2—downstream</td>
</tr>
<tr>
<td>SF_ID</td>
<td>This is always 0 because the SBC does not support DOCSIS.</td>
</tr>
<tr>
<td>Total_Bandwidth (attribute ID 253)</td>
<td>The total bandwidth being used by the streams described in a QoS_Commit message.</td>
</tr>
<tr>
<td>CCC_ID</td>
<td>The local CCC ID for a call. The attribute is included if CC tapping is being done on the call.</td>
</tr>
</tbody>
</table>

The following attributes are not included in the QoS_Commit message:
- QoS_Descriptor
- Media_Session_Desc (attribute ID 254)

Table 58-6 details the Call_Disconnect message attributes. This message is generated by the SBC when a two-way media flow is terminated. This message immediately precedes the QoS_Release and Signaling_Stop messages, and is sent only after the Call_Answer message that has been sent previously.

**Table 58-6 Call_Disconnect Message Attributes**

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>EM_Header</td>
<td>Common header attribute.</td>
</tr>
<tr>
<td>Call_Termination_Cause</td>
<td>Reason for termination of call.</td>
</tr>
</tbody>
</table>
Table 58-7 details the QoS_Release message attributes. This message is generated by the SBC when the reserved bandwidth is released.

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>EM_Header</td>
<td>Common header attribute containing timestamp and BCID.</td>
</tr>
<tr>
<td>Flow_Direction</td>
<td>1—upstream</td>
</tr>
<tr>
<td></td>
<td>2—downstream</td>
</tr>
<tr>
<td>SF_ID</td>
<td>A DOCSIS-specific attribute, Service Flow ID, generated by the CMTS in a</td>
</tr>
<tr>
<td></td>
<td>PacketCable architecture. Because the SBC does not support DOCSIS, this</td>
</tr>
<tr>
<td></td>
<td>attribute is always set to 0.</td>
</tr>
<tr>
<td>CCC_ID</td>
<td>The local CCC ID for a call. The attribute is included if CC tapping is</td>
</tr>
<tr>
<td></td>
<td>being done on the call.</td>
</tr>
</tbody>
</table>

Note

The Media_Session_Desc (attribute ID 254) attribute is not sent with the QoS_Release message.

Table 58-8 details the Signaling_Stop message attributes. This message is sent during the following events:

- A terminating signalling request, for example, a SIP BYE, from the party terminating the call is acknowledged by the SBC.
- When the terminating signalling request for the party not terminating the call is sent by the SBC, and acknowledged by that party.

Note

The Signaling_Stop message is not sent if the Signaling_Start message for this call is not sent.

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>EM_Header</td>
<td>The header attribute that must be first in the message.</td>
</tr>
<tr>
<td>Related_Call_Billing_Correlation_ID</td>
<td>The BCID of the other leg. For example, if BCID is the caller, the attribute is for the callee.</td>
</tr>
<tr>
<td>Call_Termination_Cause</td>
<td>The reason the call was terminated.</td>
</tr>
</tbody>
</table>

Note

The FEID attribute of the Signaling_Stop message is not included.

Table 58-9 details the Surveillance_Stop message attributes. This message is sent by SIG to indicate the end of IRI or CC tapping or both. This message means the call has ended.
### Table 58-9  Surveillance_Stop Message Attributes

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>EM_Header</td>
<td>Common header attribute containing the timestamp and BCID.</td>
</tr>
</tbody>
</table>
| Surveillance_Stop_Type  | Always included.  
   1—End of all surveillance.  
   2—End of only CC tapping. |
| Surveillance_Stop_Destination | Always included.  
   1—Surveillance_Stop applies to local surveillance only. The value 1 is not used by the SBC.  
   2—Surveillance_Stop is applicable to both local and remote surveillance.  
   3—Surveillance_Stop is applicable only to remote surveillance. |

**Note**

The Electronic_Surveillance_Indication attribute is not included in the Surveillance_Stop message.

Table 58-10 details the Media_Report message attributes. The message is specific to a flow. Therefore, if more than one flow is created at the same time, multiple event messages are sent, one per flow.

A Media_Report message is sent during the following events, when a flow is created, modified, and released:

- A flow is considered Created when the gate bandwidth for the flow is committed. A QoS_Commit message is also sent at the same time.
- A flow is considered Modified when the flow is renegotiated.
- A flow is considered Released when the gate bandwidth for the flow is released. A Qos_Release message is also sent at the same time.

### Table 58-10  Media_Report Message Attributes

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>EM_Header</td>
<td>Common header attribute containing the timestamp and BCID.</td>
</tr>
<tr>
<td>CCC_ID</td>
<td>The local CCCID for a call. Included if CC tapping is being done on the call.</td>
</tr>
<tr>
<td>SDP_Upstream</td>
<td>The upstream SDP for the flow, SDP corresponding to flow in direction of caller, on side indicated by Flow_Direction, is always included.</td>
</tr>
<tr>
<td>SDP_Downstream</td>
<td>The downstream SDP for the flow, SDP corresponding to flow in direction of callee, on side indicated by Flow_Direction, is always included.</td>
</tr>
</tbody>
</table>
Table 58-10  Media_Report Message Attributes (continued)

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channel_State</td>
<td>Always included.</td>
</tr>
<tr>
<td></td>
<td>1—Open (Flow created)</td>
</tr>
<tr>
<td></td>
<td>2—Change (Flow modified)</td>
</tr>
<tr>
<td></td>
<td>3—Close (Flow released)</td>
</tr>
<tr>
<td>Flow_Direction</td>
<td>Always included. Specifies if the device is acting on behalf of an</td>
</tr>
<tr>
<td></td>
<td>originating part or terminating part of a call at the time the message</td>
</tr>
<tr>
<td></td>
<td>is generated.</td>
</tr>
<tr>
<td></td>
<td>1—upstream (Caller side)</td>
</tr>
<tr>
<td></td>
<td>2—downstream (Callee side)</td>
</tr>
</tbody>
</table>

Table 58-11 details the Redirection message attributes. This message is sent by the SBC when a call has been transferred either due to a 3XX redirect response or a SIP REFER request.

Table 58-11  Redirection Message Attributes

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>EM_Header</td>
<td>Common header attribute containing the timestamp and BCID.</td>
</tr>
<tr>
<td>Related_Call_Billing_Correlator</td>
<td>Always included. The BCID used previously for the old branch of a call.</td>
</tr>
<tr>
<td>Redirected_From_Party_Number</td>
<td>Always included. The number of the party a call is being transferred from or forwarded from.</td>
</tr>
<tr>
<td>Redirected_To_Party_Number</td>
<td>Always included. The number of the party a call is being transferred to or forwarded to.</td>
</tr>
<tr>
<td>Carrier_Identification_Code</td>
<td>The Carrier Identification Code associated with a call.</td>
</tr>
</tbody>
</table>

CC Interface

The PacketCable 1.5 standard specifications contain the packet header format for replicated voice content packets.

Figure 58-3 shows a replicated packet. The first three rows of the packet are the outer Layer2, Layer3, and Layer4 information. This information consists of destination IP and UDP port of the Mediation Device, and the source IP and UDP port of the Cisco ASR 1000 series router. The fourth row of the packet is the CCC ID that is used to correlate the signaling and media information. The last four rows of the packet are the original media packet that is being TAPed. It starts from Layer 3 IP, and is followed by UDP, RTP, and media payload.
Restrictions for Implementing CALEA IRI Interface Support

The following restrictions and limitations are applicable to CALEA IRI Interface Support feature implementing:

- Only one mediation device IP address is supported.
- The IPv6 address pertaining to the mediation device is not supported. Only IPv4 address in the global routing space is supported for mediation device. The IPv4 address should not be associated to any virtual routing and forwarding (VRF).
- The mediation device’s IP address must be accessible from the Cisco ASR 1000 series router global routing space. CISCO-TAP2-MIB does not allow mediation device IP address to be in a VRF.
- The Cisco ASR 1000 series router does not support the CLIs of the Cisco BTS 10200 Softswitch and the Cisco PGW 2200 Softswitch for warrant configuration.
- The PacketCable 2.0 standard specification for Electronic Surveillance is not supported.
- LI using the SIP P-DCS-LAES header is not supported.
- Tap is not applied to the existing calls.
- The IPv6 Media Addresses cannot be intercepted in a VRF, but can be intercepted in a global routing space. However, IPv4 Media Addresses can be intercepted both in the global routing space and the VRF.
Implementing CALEA IRI Interface Support

The following sections explain how to configure the CALEA IRI Interface Support feature:

- Configuring the SBC for CALEA IRI Interface Support, page 58-16
- Configuring VoIP LI SNMP, page 58-13
- Configuring the SBC for CALEA IRI Interface Support, page 58-16

Configuring LI

To see the SNMPv3 and SNMP View configuration information pertaining to the LI TAP definitions, see the How to Configure Lawful Intercept section in the Cisco IOS and NX-OS Software Lawful Intercept Architecture feature guide at:


Use the following commands provided in the Cisco IOS and NX-OS Software Lawful Intercept Architecture feature guide to configure LI:

- `snmp-server view view-name MIB-name included` — Defines an SNMPv2 MIB view, and includes a MIB family in the view.
- `snmp-server group group-name v3 auth read view-name write view-name` — Defines a read and write view for a group using the User Security Model (SNMPv3) and the authNoPriv Security Level.
- `snmp-server user user-name group-name v3 auth md5 auth-password` — Defines an authentication password for a user by using the HMAC MD5 algorithm for authentication and V3 security model.

The following example shows how to enable the mediation device to access the lawful intercept MIBs. It creates an SNMP view (tapV) that includes three LI MIBs (CISCO-VoIp-Tap-MIB, CISCO-TAP2-MIB, and CISCO-IP-TAP-MIB). It also creates a user group that has read, write, and notify access to MIBs in the tapV view.

```
snmp-server view tapV ciscoVoIpTapMIB included
snmp-server view tapV ciscoIpTapMIB included
snmp-server view tapV ciscoTap2MIB included
snmp-server group tapGrp v3 auth read tapV write tapV notify tapV
snmp-server user li tapGrp v3 auth md5 cisco
snmp-server community public
```

Configuring VoIP LI SNMP

SNMP provisioning is done using the SNMP research tools available for Sun workstations. However, you can use any tool that uses the SNMPv3 protocol.

The `setany` commands listed here are executed using the SNMP application. Note that these commands are not Cisco IOS CLI commands. It is assumed that SNMP has been configured on your routing device. A secure K9 image is required for the MIBs to work.
Implementing CALEA IRI Interface Support

There are four parts to the following example:

- Adding the Mediation Device Information
- Adding the VoIP User Warrant
- Retrieving the Mediation Device and VoIP User Warrant Information
- Removing the VoIP User Warrant and Mediation Device Information

Adding the Mediation Device Information

Perform the following steps to add the mediation device information:

**Step 1** Configure the mediation device IP, RADIUS receiving port, transport type, and shared RADIUS Key to receive Voice signaling information from the SBC through the PacketCable1.5 Event Messages.

The following example shows how to create the TAP2 MD Row for IRI, with an IP address of 101.10.7.61, UDP port of 1813, and RADIUS key of "cisco":

```bash
setany -v3 172.18.37.151 licTap2MediationStatus.1 -i 5
setany -v3 172.18.37.151 licTap2MediationTimeout.1 -o "07 da 05 08 0e 3b 37 06"
setany -v3 172.18.37.151 licTap2MediationDestAddressType.1 -i 1
setany -v3 172.18.37.151 licTap2MediationTransport.1 -i 6
setany -v3 172.18.37.151 licTap2MediationRadiusKey.1 -o "63 69 73 63 6f"
setany -v3 172.18.37.151 licTap2MediationSrcInterface.1 -i 0
setany -v3 172.18.37.151 licTap2MediationDscp.1 -i 0
setany -v3 172.18.37.151 licTap2MediationDestAddress.1 -o "65 0a 07 3d"
setany -v3 172.18.37.151 licTap2MediationDestPort.1 -g 1813
setany -v3 172.18.37.151 licTap2MediationStatus.1 -i 1
```

**Step 2** Configure the mediation device IP, Call Content (CC) receiving port, and transport type to receive Voice CC from the SBC.

The following example shows how to create the TAP2 Mediation Device Row for a CC, with an IP address of 101.10.7.61, and UDP port of 45000:

```bash
setany -v3 172.18.37.151 licTap2MediationStatus.2 -i 5
setany -v3 172.18.37.151 licTap2MediationDestAddressType.2 -i 1
setany -v3 172.18.37.151 licTap2MediationTimeout.2 -o "07 da 05 08 0e 3b 37 06"
setany -v3 172.18.37.151 licTap2MediationTransport.2 -i 1
setany -v3 172.18.37.151 licTap2MediationSrcInterface.2 -i 0
setany -v3 172.18.37.151 licTap2MediationDscp.2 -i 0
setany -v3 172.18.37.151 licTap2MediationDestAddress.2 -o "65 0a 07 3d"
setany -v3 172.18.37.151 licTap2MediationDestPort.2 -g 45000
setany -v3 172.18.37.151 licTap2MediationStatus.2 -i 1
```

Adding the VoIP User Warrant

Perform the following steps to add the VoIP user warrant:

**Step 1** Configure the VoIP user warrant.

The following example shows how to create the VoIP TAP SNMP Row with a matching username for "712020" and type "Intercept":

```bash
setany -v3 172.18.37.151 cvoiptapStreamRowStatus.1.1 -i 5
setany -v3 172.18.37.151 cvoiptapStreamId.1.1 -o "72 72 2d 31"
setany -v3 172.18.37.151 cvoiptapStreamType.1.1 -i 4
setany -v3 172.18.37.151 cvoiptapStreamMatch.1.1 -o "37 31 32 30 32 30"
```
Chapter 58  CALEA IRI Interface Support

Implementing CALEA IRI Interface Support

Step 2

The following example shows how to configure an associated generic stream for VoIP, and enable generic stream:

```
setany -v3 172.18.37.151 li cvoiptapStreamMatchType.1.1 -i 1
setany -v3 172.18.37.151 li cvoiptapStreamCCMediationDevice.1.1 -i 2
setany -v3 172.18.37.151 li cvoiptapStreamRowStatus.1.1 -i 1
```

Step 2

The following example shows how to configure an associated generic stream for VoIP, and enable generic stream:

```
setany -v3 172.18.37.151 li cTap2StreamStatus.1.1 -i 5
setany -v3 172.18.37.151 li cTap2StreamType.1.1 -i 6
setany -v3 172.18.37.151 li cTap2StreamInterceptEnable.1.1 -i 1
setany -v3 172.18.37.151 li cTap2StreamStatus.1.1 -i 1
```

Retrieving the Mediation Device and VoIP User Warrant Information

Perform the following steps to retrieve the mediation device and VoIP user warrant information:

Step 1

The following example shows how to retrieve the MD TAP2 SNMP row:

```
getmany -v3 172.18.37.151 li ciscoTap2MIB
```

SNMP GET MANY for the configured values

- `cTap2MediationCapabilities.0 = ipV4SrcInterface(0), udp(2), radius(7)`
- `cTap2MediationDestAddressType.1 = ipv4(1)`
- `cTap2MediationDestAddressType.2 = ipv4(1)`
- `cTap2MediationDestAddress.1 = 65 0a 07 3d`
- `cTap2MediationDestAddress.2 = 65 0a 07 3d`
- `cTap2MediationDestPort.1 = 1813`
- `cTap2MediationDestPort.2 = 45000`
- `cTap2MediationSrcInterface.1 = 0`
- `cTap2MediationSrcInterface.2 = 0`
- `cTap2MediationRtcpPort.1 = 0`
- `cTap2MediationRtcpPort.2 = 0`
- `cTap2MediationDscp.1 = 0`
- `cTap2MediationDscp.2 = 0`
- `cTap2MediationDataType.1 = 0`
- `cTap2MediationDataType.2 = 0`
- `cTap2MediationRetransmitType.1 = 0`
- `cTap2MediationRetransmitType.2 = 0`
- `cTap2MediationTimeout.1 = 07 da 05 08 0e 3b 37 06`
- `cTap2MediationTimeout.2 = 07 da 05 08 0e 3b 37 06`
- `cTap2MediationTransport.1 = radius(6)`
- `cTap2MediationTransport.2 = udp(1)`
- `cTap2MediationNotificationEnable.1 = true(1)`
- `cTap2MediationNotificationEnable.2 = true(1)`
- `cTap2MediationStatus.1 = active(1)`
- `cTap2MediationStatus.2 = active(1)`
- `cTap2MediationRadiusKey.1 = cisco`
- `cTap2MediationRadiusKey.2 = cisco`

```
cTap2StreamType.1.1 = voip(6)
cTap2StreamInterceptEnable.1.1 = true(1)
cTap2StreamInterceptedPackets.1.1 = 0
```

Step 2

The following example shows how to retrieve the VoIP TAP SNMP row:

```
getmany -v3 172.18.37.151 li ciscoVoIpTapMIB
```
Removing the VoIP User Warrant and Mediation Device Information

Perform the following steps to remove the VoIP user warrant and mediation device information:

---

**Step 1**  Disable and delete the generic stream, and delete the VoIP User TAP row:

```bash
setany -v3 172.18.37.151 licTap2StreamInterceptEnable.1.1 -i 2
```

```bash
setany -v3 172.18.37.151 licvoiptapStreamRowStatus.1.1 -i 6
```

**Step 2**  Remove the mediation device RADIUS receiving port:

```bash
setany -v3 172.18.37.151 licTap2MediationStatus.1 -i 6
```

**Step 3**  Remove the MD CC receiving Port:

```bash
setany -v3 172.18.37.151 licTap2MediationStatus.2 -i 6
```

---

Configuring the SBC for CALEA IRI Interface Support

This section details the steps involved in overriding the default match-order.

**SUMMARY STEPS**

1. configure terminal
2. sbc sbc-name
3. sbe
4. adjacency sip | h323 adjacency-name
5. warrant match-order [source | destination | diverted-by]
6. exit

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>configure terminal</td>
<td>Enables global configuration mode.</td>
</tr>
</tbody>
</table>

**Example:**

- `Router# configure terminal`

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>sbc sbc-name</td>
<td>Enters the mode of an SBC service.</td>
</tr>
</tbody>
</table>

**Example:**

- `Router(config)# sbc mySbc`

  - Use the `sbc-name` argument to define the name of the service.
## Implementing CALEA IRI Interface Support

### Chapter 58      CALEA IRI Interface Support

The following example shows how to configure the SBC to override the default match-order:

```plaintext
configure terminal
sbc mySBC
sbe
adjacency sip adj1
warrant match-order source destination diverted-by
```

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 3</strong></td>
<td>sbe</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc)# sbe</td>
</tr>
<tr>
<td></td>
<td>Enters the mode of a signaling border element (SBE) entity within an SBC service.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>adjacency sip</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe)# adjacency sip sipadj</td>
</tr>
<tr>
<td></td>
<td>Enters the mode of an SBE SIP or H.323 adjacency.</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>warrant match-order [source</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-adj-sip)# warrant match-order source destination diverted-by</td>
</tr>
<tr>
<td></td>
<td>Configures the lawful enforcement warrant information in an SIP or H.323 adjacency, and specifies the order of fields used to match the warrant.</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>exit</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config-sbc-sbe-adj-sip)# exit</td>
</tr>
<tr>
<td></td>
<td>Exits the adjacency mode to the SBE mode.</td>
</tr>
</tbody>
</table>

Note: The H.323 adjacency does not support the `diverted-by` keyword.
Implementing CALEA IRI Interface Support
H.248 Border Access Controller Support

H.248 is a media gateway control protocol that enables Switched Circuit Network (SCN) to transmit voice traffic over IP. The H.248 protocol specifies master-slave architecture for decomposed gateways. In master-slave architecture, the Media Gateway Controller (MGC) is the master server and media gateways are the slave clients that behave as simple switches. One MGC can serve multiple media gateways. The H.248 protocol enables the creation, modification, and deletion of media streams across a media gateway, including the capability to negotiate the media formats to be used.

Feature History for H.248 Border Access Controller Support

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE Release 3.7</td>
<td>This feature was introduced on the Cisco ASR 1000 Series Routers.</td>
</tr>
</tbody>
</table>

Contents

This chapter contains the following sections:

- Support for the H.248 Border Access Controller, page 59-1
- Restrictions for H.248 BAC Support, page 59-3
- Prerequisites for Configuring H.248 BAC Support, page 59-4
- Configuring H.248 BAC Support, page 59-4
- Configuration Example for H.248 BAC Support, page 59-7

Support for the H.248 Border Access Controller

The session border controller (SBC) supports the H.248 Border Access Controller (BAC) feature. This feature protects the core network (with Integrated Access Devices [IADs]) from heartbeat flooding and register flooding. The BAC can terminate heartbeat from the H.248 IADs, initiate heartbeat towards IADs, and limit the register rate from IADs to the core network. The BAC hides the core MGC network topology from the IAD access adjacency, and supports media forwarding. The BAC is placed at the edge of the core network. Figure 59-1 illustrates the H.248 BAC network topology.
Support for the H.248 Border Access Controller Support

Figure 59-1   H.248 BAC Topology

```
Access Network 1

Global VRF Cisco

Adj acc1

Adj core1

Access Network 2

Global VRF Cisco

BAC

Core Network 1

Core Network 2

Global VRF Cisco

Adj acc2

Adj core2
```

---


59-2

OL-19820-15
The H.248 BAC supports the following functionalities:

- Termination of heartbeats from an access adjacency
  The BAC has two adjacencies: access adjacency and core adjacency. Only one-to-one mapping is allowed between an access adjacency and a core adjacency. The IADs and the H.248 terminal devices reside on the access adjacency. The Access Gateway Control Function (AGCF) and Media Gateway Control Function (MGCF) reside on the core adjacency. The H.248 terminal devices on the access adjacency periodically send heartbeats to the AGCF through the BAC. To decrease the impact of heartbeats on the performance of core adjacency devices such as AGCF or MGCF, the BAC sends its response to the heartbeats from the access adjacency and does not transit to the core adjacency. Therefore, the BAC can terminate heartbeats from the access adjacency.

- Topology hiding
  The BAC can modify the signaling address and the media address of the IADs and the AGCF. If it modifies these addresses, the peer will not know the original IP address of the corresponding IAD.

- Attack detection and protection
  The BAC can detect whether a signal message is from a valid or invalid H.248 terminal device. If the signal message is from an invalid H.248 terminal device, it is discarded.

- Media anchoring and forwarding
  The BAC can translate a media address according to the required configuration. When H.248 terminal devices reside in the same network, media will not flow through the core network. When media bypass is enabled, media does not anchor on the BAC.

- Signaling trace and debug
  The BAC can supply different debug levels for H.248 signaling.

### Restrictions for H.248 BAC Support

Following are the restrictions pertaining to the H.248 BAC Support feature:

- Multiple H.248 transactions in one H.248 packet are not supported.
- H.248 signaling interworking with SIP calls or H.323 calls is not supported.
- Multiple streams in local descriptors and remote descriptors are not supported.
- Memory and CPU throttle are not supported.
- Auto media bypass is not supported.
- IPv6 is not supported.
- The BAC cannot operate in the DBE mode.
- The BAC supports only the H.248 text format (long and short) message type, and not the binary format.
Prerequisites for Configuring H.248 BAC Support

The SBC must be activated before configuring the H.248 BAC Support feature. Perform the following procedure to activate the SBC:

**SUMMARY STEPS**

1. `configure terminal`
2. `sbc sbc-name`
3. `sbe`
4. `activate`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enter the global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc sbc-name</td>
<td>Create the SBC service on the Cisco Unified Border Element (SP Edition) and enter the SBC configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# sbc mySbc</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> sbe</td>
<td>Enter the mode of the signaling border element (SBE) function of the SBC.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc)# sbe</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> activate</td>
<td>Activate the SBC service.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-sbc-sbe)# activate</td>
<td></td>
</tr>
</tbody>
</table>

Configuring H.248 BAC Support

Perform the following procedure to configure the H.248 BAC Support feature:

**SUMMARY STEPS**

1. `configure terminal`
2. `sbc h248 bac`
3. `media-address ipv4 ipv4-address realm realm-number vrf vrf-name`
4. `port-range port-range`
5. `adjacency h248 [core core-adjacency name]`
6. **control-address ipv4 ipv4-address [port port number | port-range minimum-port number maximum-port number]**

7. **remote-address ipv4 ipv4-address port port number**

8. **realm realm-number**

9. **attach**

10. **exit**

11. **adjacency h248 {access access-adjacency name}**

12. **control-address ipv4 ipv4-address {port port number}**

13. **audit interval idle time**

14. **heart-beat terminate terminate-interval**

15. **domain-name domain-name**

16. **core-adj core adjacency-name**

17. **realm realm-number**

18. **attach**

---

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> Configure terminal</td>
<td>Enters the global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> sbc h248 bac</td>
<td>Configures the SBC H.248 BAC.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-h248-bac)# sbc h248 bac</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> media-address ipv4 ipv4-address realm realm-number vrf vrf-name</td>
<td>Adds an IPv4 address to the set of addresses that the BAC can use as a local media address.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-h248-bac)# media-address ipv4 8.8.8.8 realm 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> port-range port range</td>
<td>Configures the port range of the media address.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-h248-bac-media-addr)# port-range 20000 30000</td>
<td>If you do not specify the port range, the default port range values of 40000 to 65535 is applied.</td>
</tr>
<tr>
<td><strong>Step 5</strong> adjacency h248 {core core-adjacency name}</td>
<td>Configures the H.248 core adjacency and enters into the core adjacency submode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config-h248-bac)# adjacency h248 core core1</td>
<td><strong>Note</strong> Multiple core adjacencies and access adjacencies can be configured on the BAC. Always configure the core adjacency before configuring its corresponding access adjacency.</td>
</tr>
</tbody>
</table>
### Command or Action

**Step 6**  
`control-address ipv4 ipv4-address (port port number) | (port-range minimum-port number maximum-port number)`  
**Purpose**  
Configures a local IPv4 H.248 signaling address of the BAC.  
**Note**  
The BAC handles two types of Message Identifiers (MIDs): domain name and IP address. If the MID of an IAD is IP address, only the port-range is configured and not the port.

**Example:**  
Router(config-h248-bac-adj)# control-address ipv4 192.168.102.222 port-range 2944 4000

**Step 7**  
`remote-address ipv4 ipv4-address port port number`  
**Purpose**  
Configures a remote IPv4 H.248 signaling address of the MGCF and the AGCF.

**Example:**  
Router(config-h248-bac-adj)# remote-address ipv4 192.168.102.14 port 2944

**Step 8**  
`realm realm-number`  
**Purpose**  
Configures an adjacency with the IP realm that belongs to the BAC.  
A realm group can contain multiple media addresses. When you configure a realm group under an adjacency, the IP address and port for the media stream of this adjacency is allocated from the media addresses in this realm group.

**Example:**  
Router(config-h248-bac-adj)# realm 1

**Step 9**  
`attach`  
**Purpose**  
Sets the BAC adjacency state to Attached.

**Example:**  
Router(config-h248-bac-adj)# attach

**Step 10**  
`Exit`  
**Purpose**  
Exits from the core adjacency submode.

**Step 11**  
`adjacency h248 (access access-adjacency name)`  
**Purpose**  
Configures the H.248 access adjacency and enters the access adjacency submode.  
**Note**  
Always configure the access adjacency after configuring its corresponding core adjacency.

**Example:**  
Router(config-h248-bac-adj)# adjacency h248 access acc1

**Step 12**  
`control-address ipv4 ipv4-address (port port number)`  
**Purpose**  
Configures a local IPv4 H.248 signaling address of the BAC.

**Example:**  
Router(config-h248-bac-adj)# control-address ipv4 172.16.104.14 port 2940

**Step 13**  
`audit {force | interval idle time}`  
**Purpose**  
Changes the audit interval in the BAC. The default value is 1 minute.

**Example:**  
Router(config-h248-bac-adj)# audit interval 300

**Step 14**  
`heart-beat terminate terminate-interval`  
**Purpose**  
Configures the time interval during which only one heartbeat request from the H.248 terminal device can pass through the BAC and the other heartbeat requests sent during this interval are terminated.

**Example:**  
Router(config-h248-bac-adj)# heart-beat terminate 0
### Configuration Example for H.248 BAC Support

The following example shows how to configure the H.248 BAC Support feature:

```
sbc h248 bac
media-address ipv4 8.8.8.8 realm 1
port-range 20000 30000
media-address ipv4 9.9.9.9 realm 2
port-range 40000 50000
adjacency h248 core core1
    control-address ipv4 192.168.102.222 port-range 2944 4000
    remote-address ipv4 192.168.102.14 port 2944
    realm 1
    attach
adjacency h248 access acc1
    control-address ipv4 172.16.104.14 port 2940
    audit-interval 300
    heart-beat terminate 0
    domain-name cisco
    core-adj core1
    realm 2
    attach
sbc sbc
sbe
activate
```

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 15**

```
domain-name domain-name
```

Example:

```
Router(config-h248-bac-adj)# domain-name cisco
```

Specifies the domain name of the BAC adjacency that replaces the domain name of the AGCF and the MGCF.

| **Step 16**

```
core-adj core adjacency-name
```

Example:

```
Router(config-h248-bac-adj)# core-adj core1
```

Binds the BAC core adjacency with its corresponding BAC access adjacency.

| **Step 17**

```
realm realm-number
```

Example:

```
Router(config-h248-bac-adj)# realm 1
```

Configures an adjacency with the IP realm that belongs to the BAC.

A realm group can contain multiple media addresses. When you configure a realm group under an adjacency, the IP address and port for media stream of this adjacency is allocated from the media addresses in this realm group.

| **Step 18**

```
attach
```

Example:

```
Router(config-h248-bac-adj)# attach
```

Sets the BAC adjacency state to Attached.
End-to-End Cisco Unified Border Element (SP Edition) Configuration Example

This section contains a complete Cisco Unified Border Element (SP Edition) configuration on the Cisco ASR 1000 Series Routers.

Router# show run

Building configuration...

Current configuration : 17580 bytes

! Last configuration change at 11:12:56 SGT Sun Nov 21 2010

! version 15.1
service timestamps debug datetime msec localtime show-timezone
service timestamps log datetime msec localtime show-timezone
service internal
no platform punt-keepalive disable-kernel-core
platform shell
!
hostname ASR1002-2
!
boot-start-marker
boot system
bootflash:asr1000rp1-adventerprisek9.BLD_V151_1_S_XE32_THROTTLE_LATEST_20101109_090050.bin
boot system bootflash:asr1000rp1-adventerprisek9.BLD_MCP_DEV_LATEST_20101109_222533.bin
boot-end-marker
!
!
vrf definition Mgmt-intf
!
address-family ipv4
exit-address-family
!
address-family ipv6
exit-address-family
!
vrf definition h323-vrf-a
description h323-vrf-a
!
address-family ipv4
exit-address-family
!
vrf definition h323-vrf-b
description h323-vrf-b
!
address-family ipv4
exit-address-family
!
vrf definition l2e-vrf-a
description VRF a-side for late-to-early
!
address-family ipv4
exit-address-family
!
vrf definition l2e-vrf-b
description VRF b-side for late-to-early
!
address-family ipv4
exit-address-family
!
vrf definition sigpinhole_customer_a
description SigPinhole-VRF-Customer-A
!
address-family ipv4
exit-address-family
!
vrf definition sigpinhole_customer_b
description SigPinhole-VRF-Customer-B
!
address-family ipv4
exit-address-family
!
vrf definition vrf_a
!
address-family ipv4
exit-address-family
!
address-family ipv6
exit-address-family
!
logging buffered 10000000
enable secret 5 $1$wVYL$r.SbA2ka.619g7ba5dHJx/
!
no aaa new-model
!
!
no process cpu extended history
no process cpu autoprobe hog
clock timezone SGT 8 0
ip source-route
!
!
!
!
ip domain name cisco.com
ip host t-mobile.com 10.0.48.236
ip host ibcf.t-mobile.com 10.0.48.236
ip host scscf.t-mobile.com 10.0.48.236
ip name-server 20.21.28.125
ip name-server vrf vrf_a 20.21.28.125
ip name-server vrf vrf_b 20.21.28.125
!
ipv6 unicast-routing
!
!
multilink bundle-name authenticated
!
!
!
!
!
!
!
!
!
!
!
redunancy
mode none
application redundancy
  group 1
    name CUBE-SP
    shutdown
    priority 255 failover threshold 100
    control GigabitEthernet0/0/1.726 protocol 1
    data GigabitEthernet0/0/2
    protocol 1
    authentication md5 key-string cisco
!
!
!
!
!
!
!
!
!
!
ip ftp username fw
ip ftp password cisco
!
class-map type inspect match-any sip-traffic-class
match protocol sip
match protocol icmp
!
policy-map type inspect private-public-policy
  class type inspect sip-traffic-class
    inspect
class class-default
!
zone security private
zone security public
zone-pair security private-public source private destination public
service-policy type inspect private-public-policy
!
!
!
!
!
!
!
interface SBC1
  ip address 10.160.90.4 255.255.255.0 secondary
  ip address 10.160.90.11 255.255.255.0 secondary
  ip address 10.160.90.12 255.255.255.0 secondary
  ip address 10.160.90.13 255.255.255.0 secondary
  ip address 10.160.90.14 255.255.255.0 secondary
  ip address 10.160.90.15 255.255.255.0 secondary
ip address 10.160.90.16 255.255.255.0 secondary
ip address 10.160.90.17 255.255.255.0 secondary
ip address 10.160.90.18 255.255.255.0 secondary
ip address 10.160.90.19 255.255.255.0 secondary
ip address 10.160.90.3 255.255.255.0 secondary
ip address 20.24.34.1 255.255.255.0
ipv6 address 2001:A401::10:160:90:1/64
ipv6 address 2001:A401::10:160:90:2/64
ipv6 address 2001:A405::20:24:34:1/64
!
interface SBC2
  ip address 10.190.6.2 255.255.255.224 secondary
  ip address 10.190.6.1 255.255.255.224
!
interface SBC3
  ip address 10.190.6.34 255.255.255.224 secondary
  ip address 10.190.6.33 255.255.255.224
!
interface SBC4
  ip address 10.190.7.66 255.255.255.224 secondary
  ip address 10.190.7.65 255.255.255.224
!
interface SBC5
  ip address 10.190.7.98 255.255.255.224 secondary
  ip address 10.190.7.97 255.255.255.224
!
interface SBC9
  ip address 9.1.1.1 255.255.255.0
!
interface SBC200
  ip address 20.24.31.1 255.255.255.0
!
interface SBC749
  ip address 20.24.49.1 255.255.255.0
!
interface GigabitEthernet0/0/0
  ip address 1.1.1.1 255.255.255.0
  zone-member security private
  negotiation auto
  cdp enable
  redundancy rii 10
!
interface GigabitEthernet0/0/1
  no ip address
  shutdown
  negotiation auto
  cdp enable
!
interface GigabitEthernet0/0/1.726
  encapsulation dot1Q 726
  ip address 20.21.26.120 255.255.255.0
!
interface GigabitEthernet0/0/2
  ip address 1.1.2.2 255.255.255.0
  zone-member security public
  negotiation auto
!
interface GigabitEthernet0/0/3
  no ip address
  shutdown
  negotiation auto
!
interface FastEthernet0/1/0
ip address 20.21.47.16 255.255.255.0 secondary
ip address 20.21.47.13 255.255.255.0
speed 100
negotiation auto
!
interface FastEthernet0/1/1
  no ip address
  shutdown
  speed 100
  negotiation auto
!
interface FastEthernet0/1/2
  no ip address
  shutdown
  speed 100
  negotiation auto
!
interface FastEthernet0/1/3
  no ip address
  shutdown
  speed 100
  negotiation auto
!
interface GigabitEthernet0
  vrf forwarding Mgmt-intf
  ip address 10.74.48.165 255.255.255.224
  negotiation auto
!
!
no ip http server
no ip http secure-server
ip route 10.74.48.151 255.255.255.255 20.21.26.1
ip route vrf Mgmt-intf 0.0.0.0 0.0.0.0 10.74.48.161
ip route vrf Mgmt-intf 0.0.0.0 0.0.0.0 10.74.28.65
ip route vrf vrf_a 0.0.0.0 0.0.0.0 20.21.27.1
ip route vrf vrf_b 0.0.0.0 0.0.0.0 20.21.26.1
!
logging esm config
cdp run
ipv6 route vrf vrf_b ::/0 2001:20:21:26:20:21:26:1
!
!
control-plane
!
!
sbc diagnostics sparse
!
!
sbc rls8
sbe
control address aaa ipv4 20.24.34.1
radius authentication
server freeRadius
  address ipv4 10.0.48.236
  mode local
  key cisco
radius accounting Codenomicon
  concurrent-requests 4000
  retry-interval 5000
  retry-limit 9
  server Codenomicon
address ipv4 10.0.48.236
port 1812
key cisco
sip body-profile PASSALL
sip parameter-profile test
parameter aaa
  action strip
sip parameter-profile testb
sip parameter-profile proxy-param
  parameter firewall
  action strip
sip parameter-profile access-param
  parameter firewall
  action add-or-replace value public-ip-address
sip header-profile h1
src-address
  header-prio 1 header-name P-Called-Party-ID
  header-prio 2 header-name P-Preferred-Identity
header Allow entry 1
  action pass
header Call-Info entry 1
  action pass
header P-Asserted-Identity entry 1
  action pass
sip header-profile h11
header Allow entry 1
  action replace-value value "ddd"
sip header-profile p-kt
header P-KT-UE-IP entry 1
  action strip
header P-KT-UE-IP entry 2
  action add-header value "${msg.rmt_ip_addr}"
sip header-profile proxy
header contact entry 1
  parameter-profile proxy-param
  action as-profile
sip header-profile access
header contact entry 1
  parameter-profile access-param
  action as-profile
sip header-profile default
blacklist
sip header-profile IMS_Access
blacklist
header P-Called-Party-ID entry 1
  action strip
sip header-profile P-Charging-Function-Address
blacklist
header P-Charging-Function-Addresses entry 1
  action add-first-header value "1.1.1.1"
sip method-profile PASS
blacklist
sip method-profile default
blacklist
sip option-profile default
blacklist
sip error-profile default
  cause rtg-no-route-found sub-cause rtg-src-adjacency status-code 604 reason "Q.850 ;cause-16 ;text="SBC: No route found based on src adjacency""
adjacency h323 H323CCM134-GK
signaling-address ipv4 20.24.34.1
signaling-port 1719
remote-address ipv4 10.0.50.134 255.255.255.255
signaling-peer gk 10.0.48.93
Appendix 1      End-to-End Cisco Unified Border Element (SP Edition) Configuration Example

tech-prefix 567
dbe-location-id 0
allow private info
trunk trusted
inbound secure
attach
adjacency h323 H323CCM134-vrfa
  vrf h323-vrf-a
  signaling-address ipv4 10.190.7.65
  remote-address ipv4 10.0.50.134 255.255.255.255
  signaling-peer 10.0.50.134
  dbe-location-id 0
  trunk trusted
  inbound secure
  attach
adjacency sip SIPP1
  signaling-address ipv4 20.24.34.1
  statistics method summary
  signaling-port 5060
  remote-address ipv4 10.0.244.81 255.255.255.255
  signaling-peer 10.0.244.81
  dbe-location-id 0
  attach
adjacency sip SIPP2
  signaling-address ipv4 20.24.34.1
  statistics method summary
  signaling-port 5060
  remote-address ipv4 10.0.244.82 255.255.255.255
  signaling-peer 10.0.244.82
  dbe-location-id 0
  attach
adjacency sip UE-RX
  inherit profile preset-access
  signaling-address ipv4 192.168.2.1
  statistics method summary
  remote-address ipv4 10.0.120.19 255.255.255.255
  signaling-peer 10.0.120.19
  dbe-location-id 0
  reg-min-expiry 200
  fast-register disable
  attach
adjacency sip adj1-o
  inherit profile preset-access
  visited network identifier ims.net
  signaling-address ipv4 192.168.2.1
  statistics method summary
  remote-address ipv4 192.168.1.1 255.255.255.255
  signaling-peer 192.168.1.1
  media bypass tag 1 a
  media bypass tag 2 b
  media bypass tag 3 c
  media bypass tag 4 d
  attach
adjacency sip adj1-t
  inherit profile preset-access
  visited network identifier ims.net
  signaling-address ipv4 192.168.130.1
  statistics method summary
  remote-address ipv4 192.168.129.1 255.255.255.255
  signaling-peer 192.168.129.1
  media bypass tag 1 a
  media bypass tag 2 b
  media bypass tag 3 c
  media bypass tag 4 d
attached
adjacency sip CCM-132
  preferred-transport tcp
  signaling-address ipv4 20.24.34.1
  statistics method summary
  signaling-port 5060
  remote-address ipv4 10.0.50.132 255.255.255.255
  signaling-peer 10.0.50.132
  dbe-location-id 0
  ping-enable
  ping-suppression ood-request
  ping-bad-rsp-codes 503
  warrant match-order destination source diverted-by
attached
adjacency sip CCM-133
  admin-domain ad1
  vrf sigpinhole_customer_a
  signaling-address ipv4 10.190.6.33
  statistics method summary
  signaling-port 5060
  remote-address ipv4 10.0.50.133 255.255.255.255
  signaling-peer 10.0.50.133
  dbe-location-id 0
  dtmf disable sip notify
attached
adjacency sip CCM-135
  admin-domain ad1
  signaling-address ipv4 20.24.34.1
  statistics method summary
  signaling-port 5060
  remote-address ipv4 10.0.50.135 255.255.255.255
  signaling-peer 10.0.50.135
  dbe-location-id 0
  dtmf disable sip info
attached
adjacency sip OpensipsV6
  group IPv6
  nat force-off
  inherit profile preset-core
  signaling-address ipv6 2001:A401::10:160:90:1
  statistics method summary
  signaling-port 7060
  remote-address ipv6 2001::216:ECFF:FE3B:40DD/128
  signaling-peer 2001:A401::33:33:36:1
  dbe-location-id 0
  registration target address 2001:A401::33:33:36:2
  header-name From passthrough
  dtmf prefer sip info
attached
adjacency sip OpenIMSCore
  inherit profile preset-core
  signaling-address ipv4 20.24.34.1
  statistics method summary
  signaling-port 4060
  remote-address ipv4 10.0.48.236 255.255.255.255
  signaling-peer 10.0.48.236
  dbe-location-id 0
  registration target address open-ims.test
  registration monitor
  header-name From passthrough
  ims pani e2
attached
adjacency sip SoftphoneV6
  group IPv6
nat force-on
inherit profile preset-access
signaling-address ipv6 2001:A401::10:160:90:1
statistics method summary
signaling-port 5060
remote-address ipv6 2001::/64
signaling-peer 2001::10:0:120:19
dbe-location-id 0
registration rewrite-register
attach
cac-policy-set 1
cac-table SRC-ADJ
cac-table scope src-adjacency
cac-table SRC-ADJ
table-type limit src-adjacency
entry 1
match-value UE-RX
caller inband-dtmf-mode always
media police strip
action cac-complete
entry 2
match-value CCM-132
codec-preference-list pref-list1
callee-privacy privacy-service always
caller-privacy privacy-service never
srtp support allow
payload-type asymmetric allowed
callee local-call-transfer allowed
srtp callee forbid
srtp callee mandate
srtp interworking allow
media police strip
action cac-complete
entry 3
match-value CCM-133
media police strip
action next-table msmbtb1
cac-table msmbtb1
table-type policy-set
entry 1
media bypass type hairpin full
media police strip
action cac-complete
complete
cac-policy-set global 1
call-policy-set 1
first-inbound-na-table natable1
first-call-routing-table da1
first-reg-routing-table REG-ROUTE-ON-SRC-ADJ
rtg-dst-address-table da1
entry 1
match-address kate string
dst-adjacency CCM-135
action complete
entry 2
match-address bob string
dst-adjacency CCM-133
action complete
entry 3
match-address 44 digits
dst-adjacency CCM-135
action complete
prefix
entry 4
Appendix 1 End-to-End Cisco Unified Border Element (SP Edition) Configuration Example

match-address 86 digits
dst-adjacency OpenIMSCore
action complete
prefix
rtg-src-adjacency-table REG-ROUTE-ON-SRC-ADJ
entry 1
match-adjacency UE-RX
dst-adjacency OpenIMSCore
action complete
entry 2
match-adjacency SoftphoneV6
dst-adjacency OpensipsV6
action complete
entry 3
match-adjacency OpenIMSCore
dst-adjacency adj1-o
action complete
na-dst-address-table natable1
entry 1
action next-table privacytb1
edit-src add-prefix 1
match-address 111 digits
entry 2
action accept
edit-src add-prefix 12345
match-address 112 digits
entry 3
action accept
edit-src add-prefix abc
match-address 113 digits
entry 4
action accept
match-address ^201[a-d]ef regex
entry 5
action accept
na-src-name-anonymous-table privacytb1
entry 1
action accept
edit-dst add-prefix 3
match-anonymous true
complete
call-policy-set 2
first-call-routing-table ROUTE-ON-DEST-NUM
rtg-dst-address-table ROUTE-ON-DEST-NUM
entry 1
match-address 1320X digits
dst-adjacency CCM-132
action complete
edit-dst del-prefix 4
prefix
complete
call-policy-set 3
first-call-routing-table table1
rtg-src-adjacency-table table1
entry 1
match-adjacency SIPP1
dst-adjacency CCM-135
action complete
entry 2
match-adjacency SIPP2
dst-adjacency CCM-133
action complete
complete
call-policy-set default 1
admin-domain ad1
description This is a description for DOMAIN1
call-policy-set inbound-na 3
call-policy-set rtg 3
! using call-policy-set outbound-na default
admin-domain ad2
description This is a description for DOMAIN2
call-policy-set inbound-na 2
call-policy-set rtg 2
call-policy-set outbound-na 2
enum 1
  req-timeout 60
  rsp-lifetime 34000
  nmr-buf-pool-size 500
  entry default
  server ipv4 10.0.120.33
  activate
network-id 29599
sip dns
  support-type sip-dns-srv
  cache lifetime 0
  cache limit 10
!
!
codec list pref-list1
codec G723 priority 1
codec PCMU priority 2
!
codec variant codec G7231L
  variant G7231L
  standard G723
  fmtp annexa=yes
  fmtp bitrate=5.3
billing
  local-address ipv4 20.24.34.1
  ldr-check 23 30
  method packetcable-em
  method xml
  packetcable-em 0 transport radius Codenomicon
  local-address ipv4 20.24.34.1
  attach
  xml 1
  cdr path usb0:Billing/
  cdr alarm minor 500000
  ldr-check 23 30
  attach
  activate
!
!
blacklist global
reason bad-address
  trigger-size 65535
reason cac-policy-rejection
  trigger-size 65535
reason spam
  trigger-size 65535
blacklist vpn sigpinhole_customer_a
  reason authentication-failure
  trigger-size 65535
reason endpoint-registration
  trigger-size 65535
  trigger-period 1 seconds
reason cac-policy-rejection
reason corrupt-message
trigger-size 65535
trigger-period 1 seconds
blacklist global ipv6 2001::10:233:113
reason authentication-failure
trigger-size 65535
trigger-period 1 seconds
reason bad-address
trigger-size 65535
trigger-period 1 seconds
reason endpoint-registration
trigger-size 65535
trigger-period 1 seconds
reason cac-policy-rejection
trigger-size 65535
trigger-period 1 seconds
reason corrupt-message
trigger-size 65535
trigger-period 1 seconds
reason spam
trigger-size 65535
trigger-period 1 seconds
!
\textit{rtp-flood-detect}
media-address ipv4 10.160.90.3
\textit{port-range} 10000 11000 voice tag CCM-132
\textit{port-range} 11001 12000 video tag CCM-135
media-address ipv6 2001:A401::10:160:90:1
\textit{port-range} 16384 32767 signaling
activate
!
!
!
!
!
!
!
!
line con 0
exec-timeout 0 0
stopbits 1
line aux 0
stopbits 1
line vty 0 4
exec-timeout 0 0
no login
!
extension data-corruption buffer truncate
!
monitor session 22 type erspan-source
description SOURCE_SESSION_FOR_Gi0/0/0
source interface Gi0/0/0
destination
erspan-id 22
ip address 10.0.100.100
origin ip address 20.21.28.72
!
!
end
SIP Compliance and Interoperability

This appendix lists examples of Session Initiation Protocol (SIP) services and features and the type of support provided by Cisco Unified Border Element (SP Edition).

SIP Features and SBC Compliance

The following table lists some examples of SIP services and features and extent of interoperability and compliance provided by Cisco Unified Border Element (SP Edition) on the unified model.

Many of the SIP services or features in the table are listed in the draft-ietf-sipping-service-examples.txt specification.

The table covers most features offered in what are considered IP Centrex offerings from local exchange carriers and PBX (Private Branch Exchange) features. The table also includes services involving some extensions to SIP, including the REFER, SUBSCRIBE, and NOTIFY methods and the Replaces and Join headers.

<table>
<thead>
<tr>
<th>SIP Service or Feature</th>
<th>Cisco Unified Border Element (SP Edition) Compliance or Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Call Hold</td>
<td>Supported.</td>
</tr>
<tr>
<td>Consultation Hold</td>
<td>Supported.</td>
</tr>
<tr>
<td>Music on Hold</td>
<td>Supported.</td>
</tr>
<tr>
<td>Call Hold with Music on Hold</td>
<td>Supported.</td>
</tr>
<tr>
<td>Find-Me</td>
<td>Passthrough only. SBC does not perform find-me function.</td>
</tr>
<tr>
<td>Call Park</td>
<td>Passthrough only. SBC does not perform call park function.</td>
</tr>
<tr>
<td>Call Forking</td>
<td>Supported.</td>
</tr>
<tr>
<td>Caller-ID</td>
<td>Passthrough only. SBC does not perform caller-ID function.</td>
</tr>
<tr>
<td>Calling Name Delivery</td>
<td>Passthrough only. SBC does not perform calling name delivery function.</td>
</tr>
<tr>
<td>Click to Dial</td>
<td>Partial (not supported with SBC between end user computer and phone).</td>
</tr>
</tbody>
</table>
### Table 1-1  SBC Compliance and Support of SIP Services and Features (continued)

<table>
<thead>
<tr>
<th>SIP Service or Feature</th>
<th>Cisco Unified Border Element (SP Edition) Compliance or Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message Waiting Indicator</td>
<td>Passthrough, depending on SIP SUBSCRIBER/NOTIFY messages</td>
</tr>
<tr>
<td>Call Forwarding - Busy</td>
<td>Partial (not supported with SBC between proxy and callee).</td>
</tr>
<tr>
<td>Call Forwarding - No Answer</td>
<td>Partial (not supported with SBC between proxy and callee).</td>
</tr>
<tr>
<td>Call Forwarding - Unconditional</td>
<td>Supported.</td>
</tr>
<tr>
<td>SIP Session Refreshment with re-INVITE</td>
<td></td>
</tr>
<tr>
<td>SIP - Specific Event Notification</td>
<td>Partial support (RFC 3265).</td>
</tr>
<tr>
<td>Transfer - Unattended</td>
<td>Supported.</td>
</tr>
<tr>
<td>Transfer - Attended</td>
<td>Supported.</td>
</tr>
<tr>
<td>Transfer - Instant Messaging</td>
<td>Supported.</td>
</tr>
<tr>
<td>3-way Conference - Third Party is Added</td>
<td>Not supported.</td>
</tr>
<tr>
<td>3-way Conference - Third Party Joins</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Single Line Extension</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Call Management (Incoming Call Screening)</td>
<td>Passthrough only. SBC does not perform Call Screening function.</td>
</tr>
<tr>
<td>Call Management (Outgoing Call Screening)</td>
<td>Passthrough only. SBC does not perform Call Screening function.</td>
</tr>
<tr>
<td>Call Pickup</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Automatic Redial</td>
<td>Not supported.</td>
</tr>
</tbody>
</table>
**XML Billing Schema**

This appendix provides a detailed description of the XML elements used in the XML billing records that the CUBE (SP) XML billing method generates, an XML billing sample file generated by the SBC, the termination codes for the XML billing records, and the XML Document Type Definition (DTD).

**XML Elements Generated by CUBE (SP)**

This section provides details of the XML elements used in the XML billing records that the CUBE (SP) XML billing method generates.

### The recordfile Element

Table 1-1 shows the attribute in the recordfile element.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sbc</td>
<td>N</td>
<td>IP address of a CUBE (SP) recording.</td>
</tr>
</tbody>
</table>

### The call Element

Table 1-2 shows the attributes in the call element.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>starttime</td>
<td>N</td>
<td>The time at which a call starts is the time at which signaling starts.</td>
</tr>
<tr>
<td>endtime</td>
<td>Y</td>
<td>The time at which a call ends is the time at which signaling ends and resources are released. This attribute is present if the call does not end, when the call detail record (CDR) is written. This is because the billing method instance is deactivated when the billingdeactivation element is present.</td>
</tr>
</tbody>
</table>
**XML Elements Generated by CUBE (SP)**

### The subscriber Element

*Table 1-3* shows the attribute in the subscriber element.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>public_id</td>
<td>N</td>
<td>The public identifier of the subscriber.</td>
</tr>
</tbody>
</table>

### The billingdeactivation Element

*Table 1-4* shows the attribute in the billingdeactivation element.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>time</td>
<td>N</td>
<td>The time at which the billing instance is deactivated.</td>
</tr>
</tbody>
</table>
The party Element

Table 1-5 shows the attributes in the party element.

Table 1-5  Attributes of the party Element

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
</table>
| type      | N        | The applicable values are:  
|           |          | • Orig—Indicates that this party is the originating endpoint of the call.  
|           |          | • Term—Indicates that this party is the terminating endpoint of the call. |
| phone     | N        | The original phone number or the SIP user name of the party. |
| domain    | Y        | The original domain name of the phone number or the SIP user name. |
| cic       | Y        | The carrier identification code of the phone number or the SIP user name. This attribute is present only at the terminating endpoint. |
| editphone | Y        | The edited phone number or the SIP user name of the party. |
| editcic   | Y        | The edited carrier identification code of the phone number or the SIP user name. This attribute is present only at the terminating endpoint. |
| sig_address | Y   | The network address of the next-hop signaling entity. The signaling messages are received from this network address and are sent to this network address. |
| sig_port  | Y        | The network port of the next-hop signaling entity. The signaling messages are received from this network port and are sent to this network port. |

The adjacency Element

Table 1-6 shows the attributes in the adjacency element.

Table 1-6  Attributes of the adjacency Element

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
</table>
| type      | N        | The applicable values are:  
|           |          | • Orig—Indicates that this adjacency is the originating adjacency of the call.  
|           |          | • Term—Indicates that this adjacency is the outgoing adjacency of the call. |
| name      | N        | The adjacency name, as configured by the administrator on the SBC. |
| account   | N        | The account name to which the originating branch or terminating branch of the call belong to, as configured by the administrator on the SBC. |
Table 1-6 Attributes of the adjacency Element

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vpn</td>
<td>Y</td>
<td>The VPN ID associated with the adjacency, if any.</td>
</tr>
<tr>
<td>mediarealm</td>
<td>Y</td>
<td>The IP realm associated with the adjacency, if any.</td>
</tr>
</tbody>
</table>

The connect Element

Table 1-7 shows the attribute in the connect element.

Table 1-7 Attributes of the connect Element

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>time</td>
<td>N</td>
<td>The time at which the call is connected, that is, when the media gate is opened.</td>
</tr>
</tbody>
</table>

The firstendrequest Element

Table 1-8 shows the attribute in the firstendrequest element.

Table 1-8 Attributes of the firstendrequest Element

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>time</td>
<td>N</td>
<td>The time at which the first BYE request is received.</td>
</tr>
</tbody>
</table>

The disconnect Element

Table 1-9 shows the attribute in the disconnect element.

Table 1-9 Attributes of the disconnect Element

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>time</td>
<td>N</td>
<td>The time at which the call is disconnected, that is, when the final BYE response is received.</td>
</tr>
<tr>
<td>reason</td>
<td>N</td>
<td>The reason for the disconnection. For more information about the various reasons for call termination, see Table 1-17.</td>
</tr>
</tbody>
</table>

The release Element

Table 1-10 shows the attribute in the release element.

Table 1-10 Attributes of the release Element

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>reason</td>
<td>N</td>
<td>The reason for not connecting to the call. For more information about the various reasons for call termination, see Table 1-17.</td>
</tr>
</tbody>
</table>
The im_stats Element

Table 1-11 shows the attributes in the im_stats element.

Table 1-11 Attributes of the im_stats Element

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>incomplete</td>
<td>Y</td>
<td>Indicates whether the msgs_from_orig and msgs_from_term statistics attributes are applicable to the entire call. This element is omitted when the value is false.</td>
</tr>
<tr>
<td>msgs_from_orig</td>
<td>N</td>
<td>The number of IM messages sent from the caller.</td>
</tr>
<tr>
<td>msgs_from_term</td>
<td>N</td>
<td>The number of IM messages sent from the callee.</td>
</tr>
</tbody>
</table>

The QoS Element

Table 1-12 shows the attributes in the Quality of Service (QoS) element.

Table 1-12 Attributes of the QoS Element

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>reservetime</td>
<td>Y</td>
<td>The time at which the QoS is reserved.</td>
</tr>
<tr>
<td>committime</td>
<td>Y</td>
<td>The time at which the QoS is committed. This information is mandatory if the QoS is committed.</td>
</tr>
<tr>
<td>releasetime</td>
<td>N</td>
<td>The time at which the QoS is released. This field value may be inaccurate in certain post-failover scenarios such as RP switch over.</td>
</tr>
</tbody>
</table>

The gate Element

The gate element contains no attributes.

The flowinfo Element

Table 1-13 shows the attribute in the flowinfo element.

Table 1-13 Attributes of the flowinfo Element

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>transport_type</td>
<td>Y</td>
<td>This attribute can have the following values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• RTP—This indicates that the media stream is using real-time transport protocol (RTP). The RTP is the default value used, if the transport_type attribute is absent from the flowinfo element.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• SRTP—This indicates that the media stream is using secure real-time transport protocol (SRTP).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• UDPTL—This indicates that the media stream is carrying T.38 over user datagram protocol transport layer (UDPTL).</td>
</tr>
</tbody>
</table>
The local Element and the remote Element

Table 1-14 shows the attribute in the local element and the remote element.

**Table 1-14 Attributes of the local Element and the remote Element**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>address</td>
<td>N</td>
<td>The IP address that sends and receives packets.</td>
</tr>
<tr>
<td>port</td>
<td>N</td>
<td>The port number that packets are sent from and received on.</td>
</tr>
<tr>
<td>transrated</td>
<td>Y</td>
<td>Indicates whether the media packets sent to this element are transrated or not.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If this attribute is absent, transrating is not provisioned. In the current implementation of the SBC, this attribute appears only in the remote element, because transrating is always performed as late as possible.</td>
</tr>
</tbody>
</table>

The sd Element

Table 1-15 shows the attribute in the sd element.

**Table 1-15 Attributes of the sd Element**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>direction</td>
<td>Y</td>
<td>This attribute can have the following values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Inbound – Indicates that the element provides inbound SDP information.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Outbound – Indicates that the element provides outbound SDP information.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If this attribute is not included, it implies that the negotiated SDP is symmetric.</td>
</tr>
</tbody>
</table>

The RTCPStats Element

The RTCPStats element contains no attributes.

The admin_domains Element

The admin_domains element contains no attributes.
The ad Element

Table 1-16 shows the attribute in the ad element.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>N</td>
<td>The name of the admin domain associated with the endpoint.</td>
</tr>
</tbody>
</table>

Sample XML Billing File

The following is an example of an XML billing file.

```xml
<?xml version="1.0"?><recordfile sbc-sig="20.24.34.1"><call starttime="1277766440306" endtime="1277766552984" duration="112678" release_side="orig" bcid="4C29B2820202020339302B303830303000000004"><party type="orig" phone="2013" domain="10.0.50.135" sig_address="10.0.50.135" sig_port="58790"/><party type="term" phone="13208011" ediphone="8011" sig_address="10.0.0.132" sig_port="5060"/><adjacency type="orig" name="CCM-135" account="" mediarealm = "sgn1"/><adjacency type="term" name="CCM-132" account="" mediarealm = "sgn1"/><connect time="1277766442516"/><firstendrequest time="1277766552976"/><disconnect time="1277766552984" reason="0"/><QoS stream_id="1" instance="0" reservetime="1277766440306" committime="1277766442516" releasetime="1277766552987"/><gate><flowinfo><local address="20.21.4.3" port="16388"/><remote address="10.0.50.135" port="26880"/><sd>m=audio 0 RTP/AVP 0 101 a=rtpmap:0 PCMU/8000 a=fmtp:101 0-15 a=ptime:20
</sd><RTCPstats>PS=5524, OS=1104800, PR=5523, OR=1104600, PD=0, OD=0, PL=0, JI=0, LA=0, PC/RPS=0, PC/RPR=0, PC/RPL=0, PC/RLI=0, PC/RLA=0</RTCPstats></flowinfo><flowinfo><local address="20.21.4.3" port="16388"/><remote address="10.0.50.132" port="24580"/><sd>m=audio 0 RTP/AVP 0 101 a=rtpmap:0 PCMU/8000 a=fmtp:101 0-15 a=ptime:20
</sd><RTCPstats>PS=5523, OS=1104600, PR=5524, OR=1104800, PD=0, OD=0, PL=0, JI=0, LA=0, PC/RPS=0, PC/RPR=0, PC/RPL=0, PC/RLI=0, PC/RLA=0</RTCPstats></flowinfo></gate></QoS></call></recordfile>
```

Termination Codes

The following table contains the codes that describe the causes for call terminations in XML billing records.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>Normal call termination (no error).</td>
</tr>
<tr>
<td>01</td>
<td>A storage resource shortage has occurred on the local device.</td>
</tr>
<tr>
<td>02</td>
<td>A storage resource shortage has occurred on a remote device controlled by the local device.</td>
</tr>
<tr>
<td>03</td>
<td>A media resource shortage has occurred.</td>
</tr>
</tbody>
</table>
### Table 1-17  Termination Codes for XML Billing Records (continued)

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>04</td>
<td>A media failure has occurred because of the failure in the underlying hardware or through management action.</td>
</tr>
<tr>
<td>05</td>
<td>A continuity test has failed.</td>
</tr>
<tr>
<td>06</td>
<td>The requested media resource is blocked or has been quiesced.</td>
</tr>
<tr>
<td>07</td>
<td>Media is in use by another call.</td>
</tr>
<tr>
<td>08</td>
<td>Media is not configured.</td>
</tr>
<tr>
<td>09</td>
<td>An error has occurred due to a configuration inconsistency.</td>
</tr>
<tr>
<td>10</td>
<td>Media is unavailable.</td>
</tr>
<tr>
<td>11</td>
<td>Media is congested.</td>
</tr>
<tr>
<td>12</td>
<td>An internal error has occurred.</td>
</tr>
<tr>
<td>13</td>
<td>No terminations are available.</td>
</tr>
<tr>
<td>14</td>
<td>An error other than a failure, resource, or bandwidth shortage has occurred in the media layers.</td>
</tr>
<tr>
<td>15</td>
<td>A request to reset a termination has failed.</td>
</tr>
<tr>
<td>16</td>
<td>An interworking error has occurred.</td>
</tr>
<tr>
<td>17</td>
<td>A security error has occurred.</td>
</tr>
<tr>
<td>18</td>
<td>This is not a valid address.</td>
</tr>
<tr>
<td>19</td>
<td>This is not a valid transit network.</td>
</tr>
<tr>
<td>20</td>
<td>There is no route available to the specified destination address.</td>
</tr>
<tr>
<td>21</td>
<td>There is no route available to the specified transit network.</td>
</tr>
<tr>
<td>22</td>
<td>This number is unavailable because the number has changed recently.</td>
</tr>
<tr>
<td>23</td>
<td>This is an unallocated number.</td>
</tr>
<tr>
<td>24</td>
<td>There is no route-to-destination address due to congestion.</td>
</tr>
<tr>
<td>25</td>
<td>There is no route-to-transit network due to congestion.</td>
</tr>
<tr>
<td>26</td>
<td>LNP call is misrouted to the exchange that does not serve the destination number.</td>
</tr>
<tr>
<td>27</td>
<td>Internal congestion has occurred.</td>
</tr>
<tr>
<td>28</td>
<td>The media capabilities requested for the call are not supported.</td>
</tr>
<tr>
<td>29</td>
<td>The maximum number of routing retries are exceeded.</td>
</tr>
<tr>
<td>30</td>
<td>The resources are unavailable for SBC.</td>
</tr>
<tr>
<td>31</td>
<td>The destination resource is incompatible with request.</td>
</tr>
<tr>
<td>32</td>
<td>This is an invalid message.</td>
</tr>
<tr>
<td>33</td>
<td>This is an unrecognized signaling message type.</td>
</tr>
<tr>
<td>34</td>
<td>Recovery on timer expiry.</td>
</tr>
<tr>
<td>35</td>
<td>Unrecognized or unimplemented signaling parameter has been passed on.</td>
</tr>
<tr>
<td>36</td>
<td>Unrecognized or unimplemented signaling parameter has been discarded.</td>
</tr>
<tr>
<td>37</td>
<td>The signaling protocol error has occurred.</td>
</tr>
<tr>
<td>38</td>
<td>This is a temporary failure.</td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>39</td>
<td>No answer.</td>
</tr>
<tr>
<td>40</td>
<td>The destination is out of order.</td>
</tr>
<tr>
<td>43</td>
<td>Unauthorized request.</td>
</tr>
<tr>
<td>44</td>
<td>Network congestion.</td>
</tr>
<tr>
<td>45</td>
<td>The request is not supported for an unspecified reason.</td>
</tr>
<tr>
<td>46</td>
<td>The specified resource is not equipped.</td>
</tr>
<tr>
<td>47</td>
<td>Call to call services.</td>
</tr>
<tr>
<td>48</td>
<td>An unspecified or miscellaneous error has occurred.</td>
</tr>
<tr>
<td>49</td>
<td>The named digit map requested by the call agent is unknown to the media gateway.</td>
</tr>
<tr>
<td>50</td>
<td>The media bandwidth is insufficient.</td>
</tr>
<tr>
<td>51</td>
<td>The routing has failed because no digits were dialed.</td>
</tr>
<tr>
<td>52</td>
<td>A subscriber has attempted to dial a number that is restricted.</td>
</tr>
<tr>
<td>53</td>
<td>QoR call to a subscriber has failed because the subscriber was not found.</td>
</tr>
<tr>
<td>55</td>
<td>Called user has rejected the call.</td>
</tr>
<tr>
<td>56</td>
<td>The call could not be routed to a subscriber because the subscriber's termination could not be located.</td>
</tr>
<tr>
<td>57</td>
<td>Called subscriber is busy even though media can be allocated to the subscriber.</td>
</tr>
<tr>
<td>64</td>
<td>A branch that was successfully audited internally following a Call Agent failover does not indicate that the call failed during the Call Agent failover.</td>
</tr>
<tr>
<td>65</td>
<td>A subscriber attempted to register for an interval that was too brief.</td>
</tr>
<tr>
<td>66</td>
<td>This request is unauthorized by proxy.</td>
</tr>
<tr>
<td>67</td>
<td>The call’s early media exceeded the time limit set by access control before the call was connected.</td>
</tr>
<tr>
<td>68</td>
<td>A glare scenario was detected, where each party in the call sent a message of the same type simultaneously.</td>
</tr>
<tr>
<td>69</td>
<td>An endpoint has attempted a renegotiation at an illegal point.</td>
</tr>
<tr>
<td>70</td>
<td>An endpoint has sent media parameters that were unparseable.</td>
</tr>
<tr>
<td>71</td>
<td>A message or one of its subcomponents was too large to process.</td>
</tr>
<tr>
<td>72</td>
<td>An endpoint indicated that a request must be redirected.</td>
</tr>
<tr>
<td>73</td>
<td>(CAC-specific) Call setup rate have exceeded a maximum limit.</td>
</tr>
<tr>
<td>74</td>
<td>(CAC-specific) Number of call updates have exceeded the maximum limit.</td>
</tr>
<tr>
<td>75</td>
<td>(CAC-specific) Number of calls have exceeded the maximum limit.</td>
</tr>
<tr>
<td>76</td>
<td>(CAC-specific) Number of media channels used have exceeded the maximum limit.</td>
</tr>
<tr>
<td>77</td>
<td>(CAC-specific) Bandwidth used have exceeded the maximum limit.</td>
</tr>
<tr>
<td>78</td>
<td>(CAC-specific) Number of registered endpoints have exceeded the maximum limit.</td>
</tr>
<tr>
<td>79</td>
<td>(CAC-specific) Rate of endpoint registrations have exceeded the maximum limit.</td>
</tr>
</tbody>
</table>
### Table 1-17 Termination Codes for XML Billing Records (continued)

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>Media could not be established because an acceptable media transport type could not be negotiated for any media stream.</td>
</tr>
<tr>
<td>81</td>
<td>Media is not yet established because of redirection. The system is retrying.</td>
</tr>
<tr>
<td>82</td>
<td>No subscriber record with the specified search keys is found.</td>
</tr>
<tr>
<td>83</td>
<td>(CAC-specific) Rate of in-call messages have exceeded the maximum limit.</td>
</tr>
<tr>
<td>84</td>
<td>(CAC-specific) Rate of out-of-call requests have exceeded the maximum limit.</td>
</tr>
<tr>
<td>85</td>
<td>Register request from endpoint was rejected because a delegate subscriber exists in subscriber database (SUBDB) with matching search keys.</td>
</tr>
<tr>
<td>86</td>
<td>(CAC-specific) Media transport settings of the call caused it to fail.</td>
</tr>
<tr>
<td>87</td>
<td>Routing failed because the route to the address is unavailable.</td>
</tr>
<tr>
<td>88</td>
<td>No acceptable codec that can be used for a call.</td>
</tr>
<tr>
<td>89</td>
<td>The number of media channels requested is greater than the maximum number the SBC supports.</td>
</tr>
<tr>
<td>90</td>
<td>An attempt to transfer the call has failed. This is used when the reason for a call is released after an attempt to transfer it to a third party has failed.</td>
</tr>
<tr>
<td>91</td>
<td>The E.164 number mapping (ENUM) processing encountered an error.</td>
</tr>
<tr>
<td>92</td>
<td>The SBC received a message with SDP parameters that were unparsable.</td>
</tr>
<tr>
<td>93</td>
<td>A subscriber signaling bearer channel is unavailable.</td>
</tr>
<tr>
<td>94</td>
<td>A subscriber media bearer channel has failed mid-call.</td>
</tr>
<tr>
<td>95</td>
<td>A subscriber media bearer channel was rejected, either during call setup or during renegotiation.</td>
</tr>
<tr>
<td>96</td>
<td>Privacy requirements could not be satisfied for the call.</td>
</tr>
<tr>
<td>97</td>
<td>A CAC-specific error code indicating that a policy disallowing the RTP for the call caused it to fail. Note: This error code is used only in internal-to-ICC, and should not be communicated to the signaling stacks.</td>
</tr>
<tr>
<td>98</td>
<td>A CAC-specific error code indicating that a policy disallowing the SRTP for the call caused it to fail. Note: This error code is used only in internal-to-ICC, and must not be communicated to the signaling stacks.</td>
</tr>
<tr>
<td>99</td>
<td>Policy disallowing the RTP or the SRTP IW for the call caused it to fail.</td>
</tr>
<tr>
<td>100</td>
<td>No media gateway (MG) that is able to support the SRTP was found for the call, causing it to fail.</td>
</tr>
<tr>
<td>101</td>
<td>SRTP processing encountered a miscellaneous error.</td>
</tr>
<tr>
<td>102</td>
<td>Call released because media packets forwarding (MPF) has detected a fatal error.</td>
</tr>
</tbody>
</table>
XML Document Type Definition

This section provides the complete XML document type definition (DTD) for the XML billing records that the XML billing method produces.

```xml
<!DOCTYPE recordfile [ 
  <!ELEMENT recordfile (call | longcall | partialcall | audit)*>  
  <!ATTLIST recordfile sbc CDATA #REQUIRED>  
  <!ELEMENT call (subscriber, billingdeactivation, party, party, adjacency, adjacency, connect?, disconnect?, QoS*)>  
  <!ATTLIST call starttime     CDATA #REQUIRED   
               endtime       CDATA #REQUIRED   
               duration      CDATA #REQUIRED   
               release_side CDATA #IMPLIED   
               bcid          CDATA #REQUIRED>  
  <!ELEMENT subscriber EMPTY>  
  <!ATTLIST subscriber public_id  CDATA #REQUIRED>  
  <!ELEMENT billingdeactivation EMPTY>  
  <!ATTLIST billingdeactivation time CDATA #REQUIRED>  
  <!ELEMENT party (ad*)>  
  <!ATTLIST party type          CDATA #REQUIRED   
               phone         CDATA #IMPLIED   
               domain        CDATA #IMPLIED   
               cic           CDATA #IMPLIED   
               editphone     CDATA #IMPLIED   
               editcic       CDATA #IMPLIED   
               sig_address   CDATA #IMPLIED   
               sig_port      CDATA #IMPLIED   
               trunk_group   CDATA #IMPLIED   
               trunk_context CDATA #IMPLIED>  
  <!ELEMENT admin_domains (ad*)>  
  <!ELEMENT ad EMPTY>  
  <!ATTLIST ad adname             CDATA #REQUIRED>  
  <!ELEMENT adjacency EMPTY>  
  <!ATTLIST adjacency type    CDATA #REQUIRED   
               name    CDATA #REQUIRED   
               account  CDATA #REQUIRED   
               vpn     CDATA #IMPLIED   
               mediarealm CDATA #IMPLIED>  
  <!ELEMENT connect EMPTY>  
  <!ELEMENT firstendrequest EMPTY>  
  <!ELEMENT disconnect EMPTY>  
  <!ATTLIST disconnect time   CDATA #REQUIRED   
               reason CDATA #REQUIRED>  
  <!ATTLIST release time CDATA #REQUIRED>  
  <!ELEMENT im_stats EMPTY>  
  <!ATTLIST im_stats incomplete CDATA #IMPLIED   
               msgs_from_orig CDATA #REQUIRED   
               msgs_from_orig CDATA #REQUIRED>  
  <!ELEMENT QoS (gate, gate*)>  
  <!ATTLIST QoS reservetime CDATA #IMPLIED   
               committime  CDATA #IMPLIED   
               releasetime CDATA #IMPLIED>  
  <!ELEMENT gate (flowinfo, flowinfo)>  
  <!ELEMENT flowinfo (local, remote, sd, RTCPStats)>  
  <!ATTLIST flowinfo transport_type CDATA #IMPLIED>  
  <!ELEMENT local EMPTY>  
  <!ATTLIST local address CDATA #REQUIRED   
               port     CDATA #REQUIRED   
               transrated (true|false) "false">  
  <!ELEMENT remote EMPTY> 
]>
```


OL-19820-15
<!ATTLIST remote address CDATA #REQUIRED
    port CDATA #REQUIRED
    transrated (true|false) "false">
<!ELEMENT sd (#PCDATA)>
<!ATTLIST sd direction CDATA #IMPLIED>
<!ELEMENT RTCPblock (#PCDATA)>
<!ELEMENT longcall (party, party)>
<!ATTLIST longcall starttime CDATA #REQUIRED
    duration CDATA #REQUIRED
    bcid CDATA #REQUIRED>
<!ELEMENT partialcall (QoS)>
<!ATTLIST partialcall bcid CDATA #REQUIRED>
<!ELEMENT audit (log*)>
<!ELEMENT log (name, value)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT value (#PCDATA)>
]>
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:1 redundancy</td>
<td>Mechanism to provide redundancy by ensuring that for each piece of hardware there is a backup that can take over non disruptively.</td>
</tr>
<tr>
<td>1:n redundancy</td>
<td>Mechanism to provide redundancy by ensuring that for each $n$ identical pieces of hardware, there is a single backup that can take over non disruptively in the case of a single failure.</td>
</tr>
<tr>
<td>AAA address</td>
<td>Authentication, authorization, accounting address. This is the IP address used when contacting billing or authentication servers. AAA performs user/endpoint authentication prior to forwarding a request to an upstream.</td>
</tr>
<tr>
<td>Call Admission Control (CAC)</td>
<td>to control DBE</td>
</tr>
<tr>
<td>Quality of service (QoS)</td>
<td></td>
</tr>
<tr>
<td>Network Address Port Translation (NAPT) binding</td>
<td></td>
</tr>
<tr>
<td>Firewall pinhole</td>
<td></td>
</tr>
<tr>
<td>Call detail record (CDR) generation for billing</td>
<td></td>
</tr>
<tr>
<td>account</td>
<td>An account represents a service relationship with a remote organization on the SBE. Each adjacency is assigned to an account, which is used to define customer-specific Call Admission Control and routing policy configuration.</td>
</tr>
<tr>
<td>admission control policy</td>
<td>A set of rules on the SBE that define system and call level restrictions.</td>
</tr>
<tr>
<td>ALG</td>
<td>Application layer gateway. A bridge for traffic between two networks. It has knowledge of, and operates at the level of, the application generating the traffic.</td>
</tr>
<tr>
<td>B2BUA</td>
<td>Back-to-back user agent. This is a piece of software that links together the signaling flows for two legs of a call, providing a bridge between them with local termination for each leg.</td>
</tr>
</tbody>
</table>
CAC  
Call Admission Control. This is the set of actions taken by a network during the set-up phase of a call event to determine whether the event should be accepted or rejected.

call policy  
An interconnected set of rules used to configure how SBC responds to new call events. It includes number analysis, routing, and CAC.

CALEA  
Communications Assistance for Law Enforcement Act. Passed in 1994, CALEA requires telecommunications carriers in the United States to modify their equipment, facilities, and services to ensure that they are able to comply with authorized electronic surveillance.

CDR  
Call detail record. The billing record for a phone call.

CE  
See PE.

codec  
Compressor/decompressor. A codec is any technology for compressing and decompressing data, typically audio or video.

control address  
IP address on the SBE or DBE used for terminating the H.248 control traffic between SBE and SBE. Also used in AAA control traffic.

COPS-PR  
Common Open Policy Service. This is an IETF standard, supplying network switches and hubs with policy rules to help maintain quality of service.

CORBA  
Common Object Request Broker Architecture. CORBA is an architecture and specification for creating, distributing, and managing distributed program objects in a network.

DoS protection  
Protects SBE from DoS (Denial of Service) attack.

DBE  
Data border element, also known as the media proxy. Represents the media-handling portion (RTP, RTCP, and so on) of the SBC. There can be only one DBE per service card. However, the DBE can be partitioned into several virtual DBEs (VDBEs). The DBE supports the following services:

- Bandwidth allocation, Call Admission Control (CAC), and Service Level Agreement (SLA) Monitoring
- Policing, marking (DSCP), and rate limiting
- RSVP proxy
- Firewall (media pinholes)
- Security functions
- NAPT traversing
- Topology hiding
- VPN aware (VPN interconnect)
- Quality monitoring and statistics gathering
DSP service control  Engages in the codec negotiation procedures and enforces policy on codecs being negotiated to control digital signal processor (DSP) service.

DiffServ  Differentiated services. A mechanism for marking IP traffic with different priorities.

DoS  Denial of service. A malicious attempt to overload a piece of hardware in some way.

DMZ  Demilitarized zone. This is a small subnetwork that sits between a trusted private network, such as a corporate LAN, and an untrusted public network, such as the public Internet.

F

firewall  A system designed to protect a computer network from unauthorized access, especially through the Internet.

H

H.248  H.248 (or Megaco) is a VoIP signaling protocol, usually used between a dumb device and a clever controller. It is similar in functionality (if not syntax) to MGCP. It is used to communicate between SBC and DBE in a distributed SBC system.

H.323  A protocol used for signaling for VoIP.

HSD  Hot software downgrade.

I

IAD  Integrated access device. An IAD is a one-box DSL voice and data solution equipment typically installed at the customer’s site.

L

Lawful intercept  Provides intercept-related information (IRI) and call content intercept (replication of the media streams).

load-related services (sharing and balancing)  SBE may also perform load balancing when it sends a message to multiple upstream or downstream servers.

location ID  Identifies the location of DBE within the network.

LSP  Label switched path. The name for a single traffic flow in MPLS.
### M

**media address**  
Pool of IP addresses on the DBE for media relay functionality. A separate pool of addresses is defined for each VPN that the DBE is attached to. All vDBEs within the DBE draw media addresses from these pools.

**media bypass**  
An SBC function allowing media to bypass DBE and flow directly between two endpoints within the same customer network or VPN.

**media transcoding device**  
A type of media gateway that can convert between media codec types in real time. SBEs sometimes include a combination of vDBE and a media transcoding device in the data path of a single call.

**megaco**  
See H.248.

**MGCP**  
Media Gateway Control Protocol. This is a VoIP signaling protocol, usually used between a dumb device and a clever controller. It is similar in functionality (if not syntax) to H.248/Megaco. It is defined in RFC 2705.

**MPLS**  
Multiprotocol Label Switching. Protocol used for network traffic flow shaping and management.

**message scrubbing for identity and address hiding**  
Hiding end-user identifying information and end-user IP-addresses by adding, removing, or modifying the identity and IP address information in the signaling headers.

### N

**NAT**  
Network Address Translator. This is a program or piece of hardware that converts an IP address from a private address to a public address in real time. It allows multiple users to share a single public IP address.

**NAT traversal**  
Detects that the endpoints are behind a NAT device and provide NAT traversal.

**NNI**  
Network to network interface. The border between two carriers.

**Number analysis**  
A set of rules to determine whether a called number is valid and, optionally, to assign a category to the call or edit the called number.

### O

**OAM**  
Operation, administration, and maintenance.

### P

**PE**  
Provider edge. This is a piece of equipment situated at the edge of a service provider’s network, typically contrasted with Customer Edge (CE) equipment.
**POTS**
Plain old telephone service. This is the standard telephone service that most homes use. It is also referred to as the PSTN.

**PSTN**
Public Switched Telephone Network. The world’s collection of interconnected voice-oriented public telephone networks.

**R**

**RADIUS**
Remote Authentication Dial-In User Service. Protocol used by SIG to connect to call accounting services or authentication services.

**routing policy**
A set of rules on the SBE to determine the next-hop VoIP signaling entity to which a signaling request should be sent. It defines whether a given called number is valid, and if so, where to send outbound signaling.

**RSIP**
Realm-Specific Internet Protocol. An IP address translation technique that is an alternative to NAT. RSIP lets an enterprise safeguard many private Internet addresses behind a single public Internet address.

**RTCP**
Real-Time Control Protocol. A protocol to carry information on the performance of RTP traffic.

**RTP**
Real-Time Protocol. This is the dominant protocol for carrying VoIP media data. It is defined in RFC 3550.

**S**

**SBE**
Signaling border element (also known as **signaling proxy**). Represents the signaling agent of the SBC to handle all call processing through SIP or H.323 protocols. There can be only one signaling agent per service card. An SBE typically controls one or more media gateways. The SBE supports the following services:

- Call Admission Control (CAC)
- Signaling scrubbing
- Security functions
- Routing
- Registration/authentication
- Identity hiding
- Topology hiding
- Protocol conversion
- Facilitate transcoding by communicating with the media gateway or media server

**SDP**
Session Description Protocol. A syntax for describing key features of media streams, including codecs, IP addresses and ports, bit rates, and other information. It is defined in RFC 2327.

**Session Control Interface (SCI)**
SCI controls the various DBE entities in a distributed mode of operation.
signaling address: IP address on the SBE for terminating VoIP signaling (that is, SIP, H.323). A signaling address may be qualified by a VPN ID (VRF name) if the SBE needs to be assigned private addresses specific to particular VPNs.

signaling protocol translation and interworking: Performs protocol translation between different signaling protocols such as SIP and H.323.

SIP: Session Initiation Protocol. A protocol used for signaling for VoIP.

SLA: Service Level Agreement. The contract between a service provider and the customer that specifies the level of service that will be provided.

SNMP: Simple Network Management Protocol. An Internet standard that defines methods for remotely managing active network components such as hubs, routers, and bridges.

SOAP: Simple Object Access Protocol. A way for a web server to call a procedure on another, physically separate web server, and get back a machine-readable result in standard XML format.

SP: Service provider.

SVI: Service virtual interface.

T


TLS: Transport Layer Security. A protocol that provides data integrity and privacy on a communications link over the Internet. It allows client/server applications to communicate and is designed to prevent eavesdropping, message forgery, and interference.

topology and infrastructure hiding: Hiding organization topology and infrastructure by removing routing information or by modifying the From/Contact information in the signaling headers.

transcoder: Technology for converting between different codecs.

U

UDP: User Datagram Protocol. This is a transport layer protocol in the TCP/IP protocol suite, used in the Internet. UDP is used at the two ends of a data transfer. It does not establish a connection or provide reliable data transfer like TCP.

UNI: User-to-Network Interface. The border between a service provider and the customer.
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDBE</td>
<td>Represents a resource partition within a DBE. A VDBE is a type of media gateway. Each VDBE can be controlled by a separate SBE using the H.248 (Megaco) protocol.</td>
</tr>
<tr>
<td>VoIP</td>
<td>Voice over IP.</td>
</tr>
<tr>
<td>VPN</td>
<td>Virtual Private Network.</td>
</tr>
<tr>
<td>VRF</td>
<td>Virtual Routing and Forwarding Instances</td>
</tr>
<tr>
<td>VoIP signaling peer</td>
<td>Peer device within the VoIP signaling network.</td>
</tr>
<tr>
<td>VoIP event</td>
<td>Significant events within the VoIP network, such as new calls, call updates, and subscriber registrations.</td>
</tr>
</tbody>
</table>