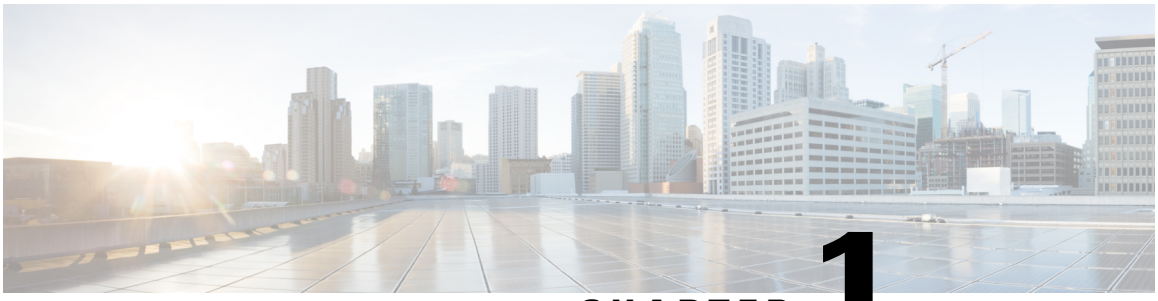
Cisco Unified Border Element (Enterprise) Fundamentals and Basic Setup Configuration Guide, Cisco IOS XE Release 3S

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS



CHAPTER 1

Overview of Cisco Unified Border Element

Cisco Unified Border Element (CUBE) is a unified communications border element, providing voice and video connectivity between the enterprise IP network and service provider network. It is similar to a voice gateway, except for the replacement of physical voice trunks with an IP connection.

Information about Cisco Unified Border Element

Cisco Unified Border Element (CUBE) is network border element that can terminate and originate signaling (H.323 and Session Initiation Protocol [SIP]), media streams (Real-Time Transport Protocol [RTP] and RTP Control Protocol [RTCP]).

Session Border controller (SBC) was used by service providers (SP) to enable full billing capabilities within VoIP networks. CUBE provides the extended functionality of interconnecting VoIP networks, especially on the enterprise side.

CUBE functionality is implemented on devices using a special IOS feature set, which allows CUBE to route a call from one VoIP dial peer to another. As VoIP dial peers can be handled by either SIP or H.323, CUBE can be used to interconnect VoIP networks of different signaling protocols. VoIP internetworking is achieved by connecting an inbound dial peer with an outbound dial peer. A standard Cisco IOS gateway without CUBE functionality cannot allow VoIP-to-VoIP connections.

Protocol internetworking is possible for the following combinations:

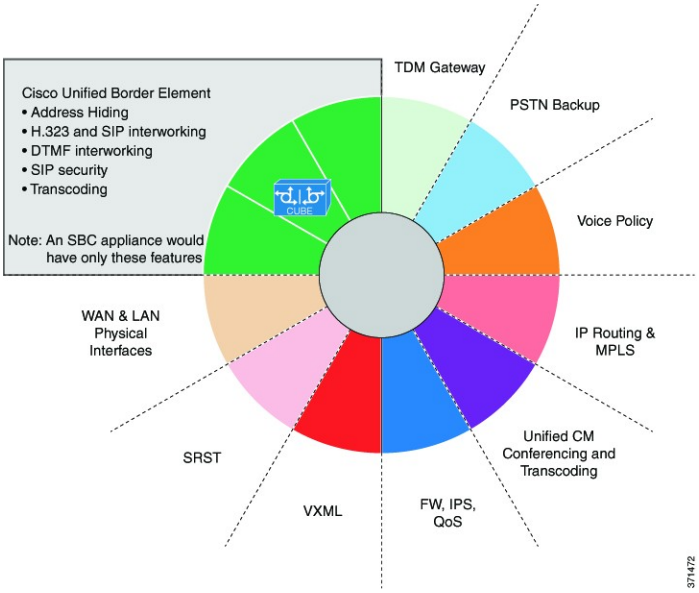
- H.323-to-SIP internetworking
- H.323-to-H.323 internetworking
- SIP-to-SIP internetworking

CUBE is used by enterprise and small and medium-sized organizations to interconnect SIP PSTN access with SIP and H.323 enterprise unified communications networks.

A CUBE interoperates with several different network elements including voice gateways, IP phones, and call-control servers in many different application environments, from advanced enterprise voice and/or video services with Cisco Unified Communications Manager or Cisco Unified Communications Manager Express, as well as simpler toll bypass and voice over IP (VoIP) transport applications. The CUBE provides organizations

with all the border controller functions integrated into the network layer to interconnect unified communications voice and video enterprise-to-service-provider architectures.

Figure 1: Cisco Unified Border Element—More than an SBC

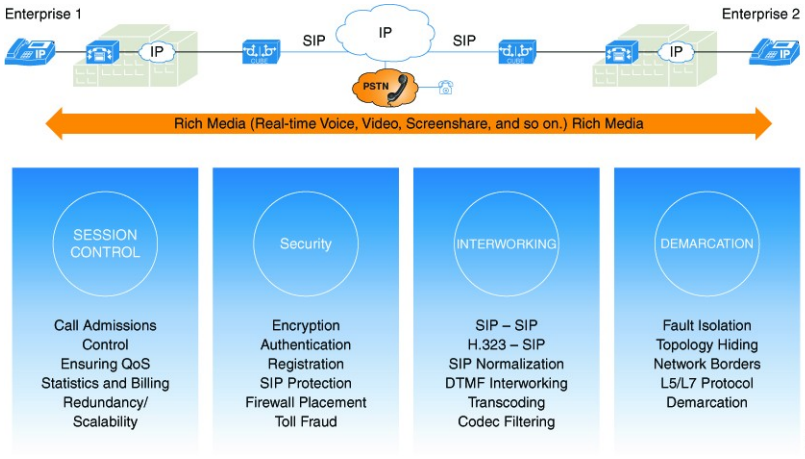


The CUBE provides a network-to-network interface point for:

- Signaling interworking—H.323 and SIP.
- Media interworking—dual-tone multifrequency (DTMF), fax, modem, and codec transcoding.
- Address and port translations—privacy and topology hiding.
- Billing and call detail record (CDR) normalization.

- Quality-of-service (QoS) and bandwidth management—QoS marking using differentiated services code point (DSCP) or type of service (ToS), bandwidth enforcement using Resource Reservation Protocol (RSVP), and codec filtering.

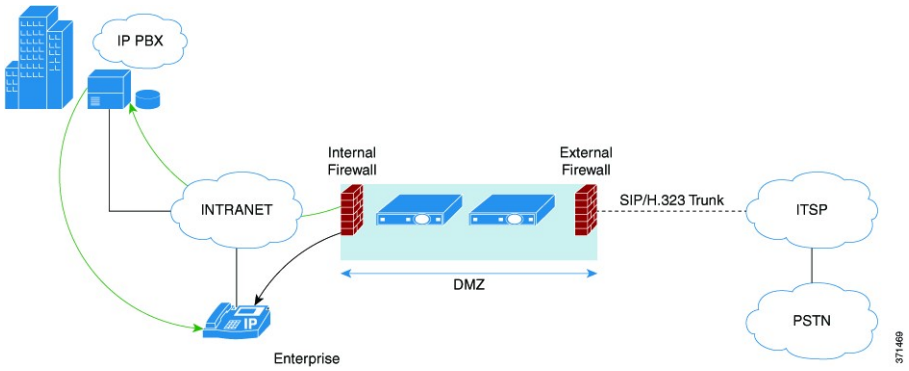
Figure 2: Why does an enterprise need the CUBE?



SIP/H.323 Trunking

The Session Initiation Protocol (SIP) is a signaling communications protocol, widely used for controlling multimedia communication sessions such as voice and video calls over Internet Protocol (IP) networks. SIP (or H.323) trunking is the use of VoIP to facilitate the connection of a private branch exchange (PBX) to the Internet. To use SIP trunking, an enterprise must have a PBX that connects to all internal end users, an Internet telephony service provider (ITSP) and a gateway that serves as the interface between the PBX and the ITSP. One of the most significant advantages of SIP trunking is its ability to combine data, voice, and video in a single line, eliminating the need for separate physical media for each mode.

Figure 3: SIP/H.323 Trunking

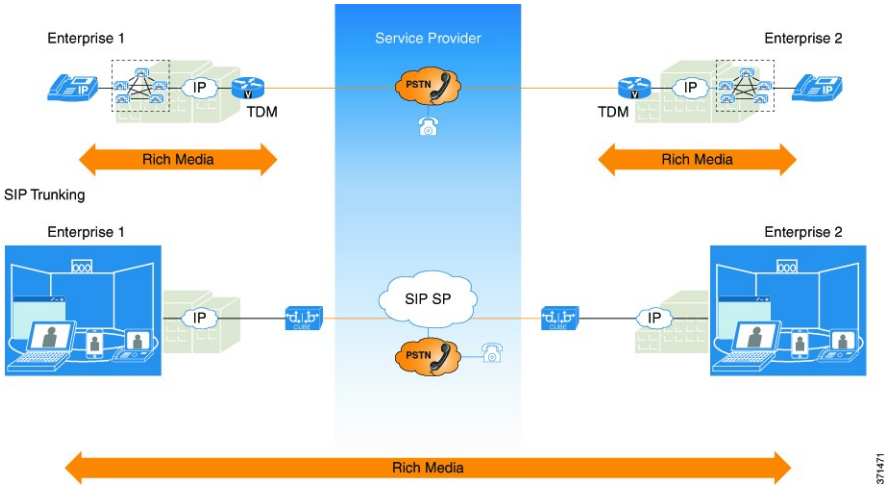


SIP trunking overcomes TDM barriers, in that it:

- Improves efficiency of interconnection between networks
- Simplifies PSTN interconnection with IP end-to-end

- Enables rich media services to employees, customers, and partners
- Carries converged voice, video, and data traffic

Figure 4: SIP Trunking overcomes TDM Barriers



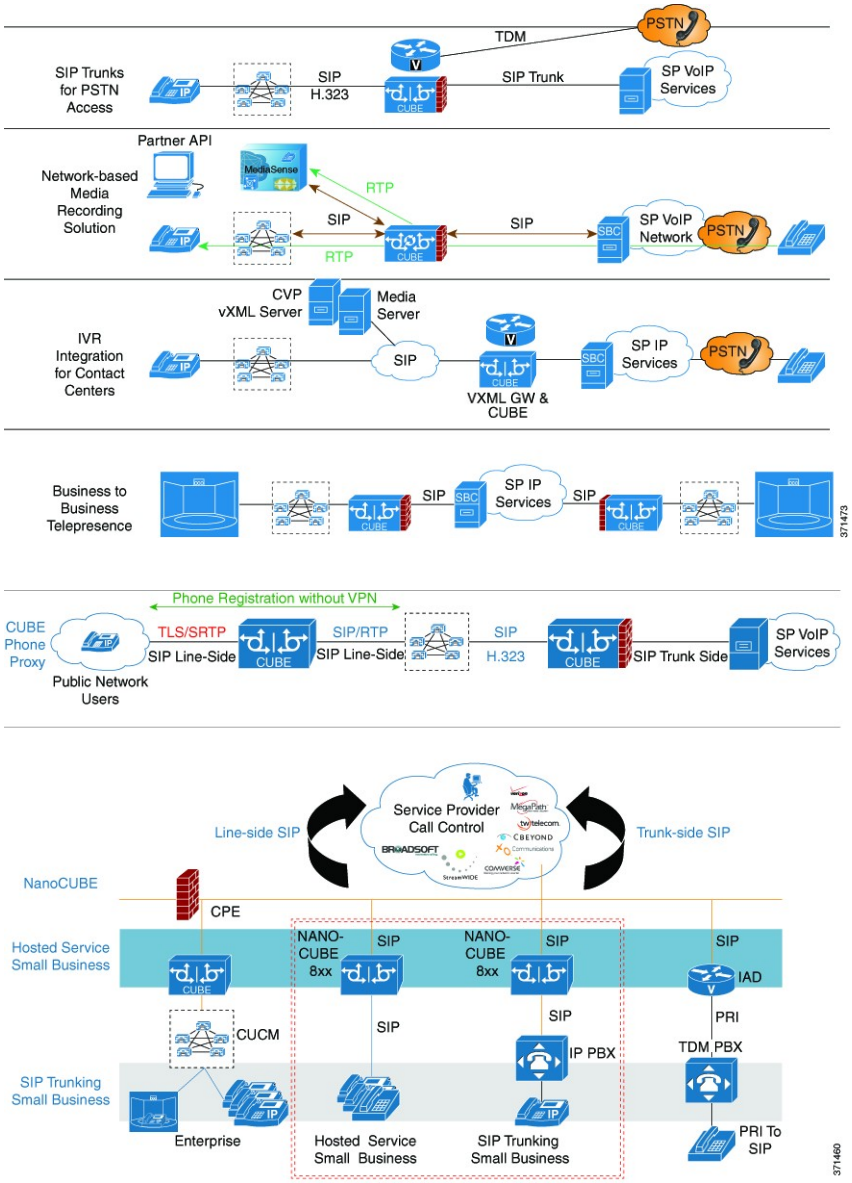
Typical Deployment Scenarios for CUBE

CUBE in an enterprise environments serve two main purposes:

- External Connections-CUBE is the demarcation point within a unified communications network and provides interconnectivity with external networks. This includes H.323 voice and video connections and SIP VoIP connections.

- Internal Connections-When used within a VoIP network, CUBE increases flexibility and interoperability between devices.

Figure 5: Typical Deployment Scenarios

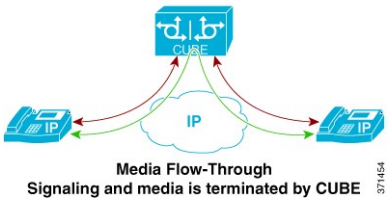


CUBE Deployment Modes

- Media flow-through—CUBE acts as a back-to-back user agent. In a media flow-through mode, between two endpoints, both signaling and media flows through the IP-to-IP Gateway (IPIP GW). The IPIP GW

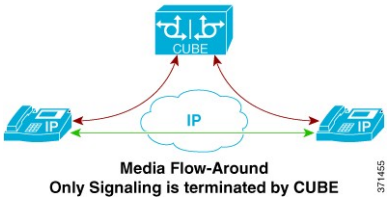
performs both signaling and media interworking between H.323/SIP IPv4 and SIP IPv6 networks. Media-flow-through mode is recommended when CUBE is used as an SBC for PSTN connectivity.

Figure 6: Media flow-through



- Media flow-around—Only signaling is terminated at CUBE. Media bypasses CUBE and flows directly between the endpoints. This mode is recommended to be used only if CUBE is deployed within an enterprise network.

Figure 7: Media flow-around



How to Configure Basic CUBE Tasks

Enabling the CUBE application on a Router

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **mode border-element license capacity** *sessions*
5. **allow-connections** *from-type* **to** *to-type*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**voice service voip** **Example:** `Device(config)# voice service voip`	Enters global VoIP configuration mode.
Step 4	**mode border-element license capacity** *sessions* **Example:** `Device(conf-voi-serv)# mode border-element license capacity 200`	Enables the set of commands used in the CUBE. • You can configure the number of licenses (capacity) to be enabled for the CUBE.
Step 5	**allow-connections** *from-type* **to** *to-type* **Example:** `Device(conf-voi-serv)# allow-connections sip to sip`	Allows connections between specific types of endpoints in a VoIP network. • The two protocols (endpoints) refer to the VoIP protocols on the two call legs.

	Command or Action	Purpose
Step 6	**end** **Example:** `Device(conf-voi-serv)# end`	Returns to privileged EXEC mode.

Configuring a Trusted IP Address List for Toll-Fraud Prevention

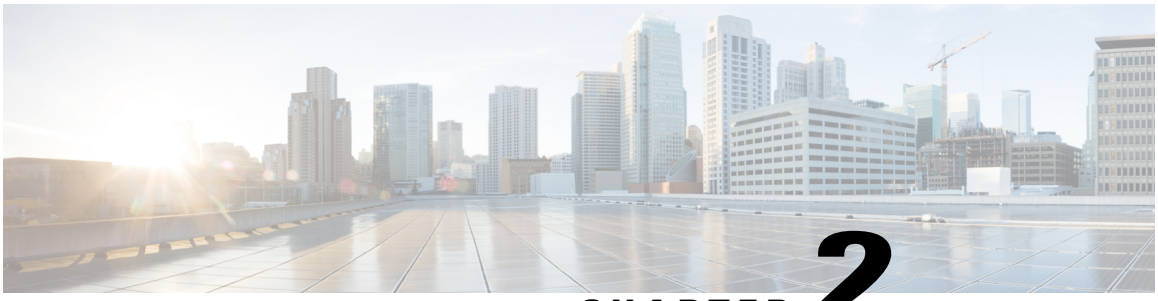
SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **ip address trusted list**
5. **ipv4** *ipv4-address* [*network-mask*]
6. **ipv6** *ipv6-address*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**voice service voip** **Example:** `Device(config)# voice service voip`	Enters voice service configuration mode.
Step 4	**ip address trusted list** **Example:** `Device(conf-voi-serv)# ip address trusted list`	Enters IP address trusted list mode and enables the addition of valid IP addresses.

	Command or Action	Purpose
Step 5	**ipv4** *ipv4-address* [*network-mask*] **Example:** `Device(cfg-iptrust-list)# ipv4 192.0.2.1`	Allows you to add up to 100 IPv4 addresses in the IP address trusted list. Duplicate IP addresses are not allowed. • The *network-mask* argument allows you to define a subnet IP address.
Step 6	**ipv6** *ipv6-address* **Example:** `Device(cfg-iptrust-list)# ipv6 2001:DB8:0:ABCD::1/48`	Allows you to add IPv6 addresses to the trusted IP address list.
Step 7	**end** **Example:** `Device(cfg-iptrust-list)# end`	Returns to privileged EXEC mode.



CHAPTER 2

Configuring SIP Bind Features

The SIP Gateway Support for the bind Command feature allows you to configure the source IP address of signaling packets and media packets.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for SIP Bind Features

The following are the prerequisites for this feature:

- Ensure the gateway has voice functionality that is configurable for Session Initiation Protocol (SIP).
- Establish a working IP network. For more information about configuring IP, refer to the *Cisco IOS IP Addressing Configuration Guide*.

- Configure VoIP. For more information about configuring VoIP, refer to the *Cisco IOS Voice Command Reference*.

Restrictions for SIP Bind Features

Although the **bind all** command is an accepted configuration, it does not appear in **show running-config** command output. Because the **bind all** command is equivalent to issuing the commands **bind source** and **bind media**, those are the commands that appear in the **show running-config** command output.

Information About SIP Bind Features

When you configure SIP on a router, the ports on all its interfaces are open by default. This makes the router vulnerable to malicious attackers who can execute toll fraud across the gateway if the router has a public IP address and a public switched telephone network (PSTN) connection. To eliminate the threat, you should bind an interface to an IP address so that only those ports are open to the outside world. In addition, you should protect any public or untrusted interface by configuring a firewall or an access control list (ACL) to prevent unwanted traffic from traversing the router.

Benefits of SIP Bind Features

The benefits of SIP Bind feature is as follows:

- SIP signaling and media paths can advertise the same source IP address on the gateway for certain applications, even if the paths used different addresses to reach the source. This eliminates confusion for firewall applications that may have taken action on source address packets before the use of binding.
- Firewalls filter messages based on variables such as the message source, the target address, and available ports. Normally a firewall opens only certain addresses or port combination to the outside world and those addresses can change dynamically. Because VoIP technology requires the use of more than one address or port combination, the **bind** command adds flexibility by assigning a gateway to a specific interface (and therefore the associated address) for the signaling or media application.
- You can obtain a predefined and separate interface for both signaling and media traffic. Once a **bind** command is in effect, the interface it limits is bound solely to that purpose. Administrators can therefore dictate the use of one network to transport the signaling and another network to transport the media. The benefits of administrator control are:
  - Administrators know the traffic that runs on specific networks, thereby making debugging easier.
  - Administrators know the capacity of the network and the target traffic, thereby making engineering and planning easier.
  - Traffic is controlled, allowing Qualtiy of Service (QoS) to be monitored.
- The **bind media** command relaxes the constraints imposed by the **bind control** and **bind all** commands, which cannot be set during an active call. The **bind media** command works with active calls.

To configure SIP Gateway Support for the bind Command, you should understand the following concepts:

Source Address

In early releases of Cisco IOS software with SIP functionality, the source address of a packet going out of the gateway was never deterministic. That is, the session protocols and VoIP layers always depended on the IP layer to give the *best local address* . The best local address was then used as the source address (the address showing where the SIP request came from) for signaling and media packets. Using this nondeterministic address occasionally caused confusion for firewall applications, because a firewall could not be configured with an exact address and would take action on several different source address packets.

However, the **bind** command allows you to configure the source IP address of signaling and media packets to a specific interface's IP address. Thus, the address that goes out on the packet is bound to the IP address of the interface specified with the **bind** command. Packets that are not destined to the bound address are discarded.

When you do not want to specify a bind address or if the interface is down, the IP layer still provides the best local address.

The Support Ability to Configure Source IP Address for Signaling and Media per SIP Trunk feature extends the global bind functionality to support the SIP signaling Transport Layer Socket (TLS) with UDP and TCP. The source address at the dial peer is the source address in all the signaling and media packets between the gateway and the remote SIP entity for calls using the dial-peer. Multiple SIP listen sockets with specific source address handle the incoming SIP traffic from each selected SIP entity. The order of preference for retrieving the SIP signalling and media source address for inbound and outbound calls is as follows:

- Bind configuration at dial peer level
- Bind configuration at global level
- Best local IP address to reach the destination

The table below describes the state of the system when the **bind** command is applied in the global or dial peer level:

Table 1: State of the System for the bind Address

Bind State	System Status
No global bind	The best local address is used in all outbound SIP messages. Only one SIP listen socket with a wildcard source address.
Global bind	Global bind address used in all outbound SIP messages. Only one SIP listen socket with global bind address.
No global bind Dial peer bind	Dial peer bind address is used in outbound SIP messages of this dial peer. The remaining SIP messages use the best local address. One SIP listen socket with a wildcard source address. Additional SIP listen socket for each different dial peer bind listening on the specific dial peer bind address.

Bind State	System Status
Global bind Dial peer bind	Dial peer bind address is used in outbound SIP messages of this dial peer. The remaining SIP messages use the global bind address. One SIP listen socket with global bind address. Additional SIP listen socket for each different dial peer bind command listening on the specific dial peer bind address.

The **bind** command performs different functions based on the state of the interface (see the table below).

Table 2: State of the Interface for the bind Command

Interface State	Result Using Bind Command
Shut down With or without active calls	TCP, TLS, and User Datagram Protocol (UDP) socket listeners are initially closed. (Socket listeners receive datagrams addressed to the socket.) Then the sockets are opened to listen to any IP address. If the outgoing gateway has the **bind** command enabled and has an active call, the call becomes a one-way call with media flowing from the outgoing gateway to the terminating gateway. The dial peer bind socket listeners of the interface are closed and the configuration turns inactive for all subsequent SIP messages.
No shut down No active calls	TCP, TLS, and UDP socket listeners are initially closed. (Socket listeners receive datagrams addressed to the socket.) Then the sockets are opened and bound to the IP address set by the **bind** command. The sockets accept packets destined for the bound address only. The dial peer bind socket listeners of the interface are reopened and the configuration turns active for all subsequent SIP messages.

Interface State	Result Using Bind Command
No shut down Active calls	TCP, TLS, and UDP socket listeners are initially closed. Then the sockets are opened to listen to any IP address. The dial peer bind socket listeners of the interface are reopened and the configuration turns active for all subsequent SIP messages.
Bound-interface IP address is removed	TCP, TLS, and UDP socket listeners are initially closed. Then the sockets are opened to listen to any address, because the IP address has been removed. This happens even when SIP was never bound to an IP address. A message stating that the IP address has been deleted from the SIP bound interface is printed. If the outgoing gateway has the **bind** command enabled and has an active call, the call becomes a one-way call with media flowing from the outgoing gateway to the terminating gateway. The dial peer bind socket listeners of the interface are closed and the configuration turns inactive for all subsequent SIP messages.
The physical cable is pulled on the bound port, or the interface layer is down	TCP, TLS, and UDP socket listeners are initially closed. Then the sockets are opened and bound to listen to any address. When the pulled cable is replaced, the result is as documented for **no shutdown** interfaces. The dial peer bind socket listeners of the interface are closed and the configuration turns inactive for all subsequent SIP messages.

Interface State	Result Using Bind Command
A bind interface is shut down, or its IP address is changed, or the physical cable is pulled while SIP calls are active	The call becomes a one-way call with media flowing in only one direction. It flows from the gateway where the change or shutdown took place, to the gateway where no change occurred. Thus, the gateway with the status change no longer receives media. The call is then disconnected, but the disconnected message is not understood by the gateway with the status change, and the call is still assumed to be active. If the bind interface is shutdown, the dial peer bind socket listeners of the interface are closed. If the IP address of the interface is changed, the socket listeners representing the bind command is opened with the available IP address of the interface and the configuration turns active for all subsequent SIP messages.
Note If there are active calls, the **bind** command does not take effect if it is issued for the first time or another **bind** command is in effect. A message reminds you that there are active calls and that the change cannot take effect.	

The **bind** command applied at the dial peer level can be modified only in the following situations:

- Dial peer bind is disabled in the supported IOS configuration options.
- Dial peer bind is removed when the bound interface is removed.
- Dial peer bind is removed when the dial peer is removed.

Voice Media Stream Processing

The SIP Gateway Support Enhancements to the bind Command feature extends the capabilities of the **bind** command by supporting a deterministic network interface for the voice media stream. Before the voice media stream addition, the **bind** command supported a deterministic network interface for control (signaling) traffic or all traffic. With the SIP Gateway Support Enhancements to the bind Command feature a finer granularity of control is achieved on the network interfaces used for voice traffic.

If multiple **bind** commands are issued in sequence--that is, if one **bind** command is configured and then another **bind** command is configured--a set interaction happens between the commands. The table below describes the expected command behavior.

Table 3: Interaction Between Previously Set and New bind Commands

Interface State	bind Command	Result Using bind Command
Without active calls	**bind all**	Generated **bind control** and **bind media** commands to override existing **bind control** and **bind media** commands.
	bind control	Overrides existing **bind control** command.
	bind media	Overrides existing **bind media** command.
With active calls	**bind all** or **bind control**	Blocks the command, and the following messages are displayed: 00:16:39: There are active calls 00:16:39: configure_sip_bind_command: The bind command change will not take effect
	bind media	Succeeds and overrides any existing **bind media**command.

The **bind all** and **bind control**commands perform different functions based on the state of the interface. The table below describes the actions performed based on the interface state.

Note The **bind all** command only applies to global level, whereas the **bind control** and **bind media** command apply to global and dial peer. The table below applies to **bind media** only if the media interface is the same as the **bind control** interface. If the two interfaces are different, media behavior is independent of the interface state.

Table 4: bind all and bind control Functions, Based on Interface State

Interface State	Result Using bind all or bind control Commands
Shut down With or without active calls	TCP, TLS, and UDP socket listeners are initially closed. (Socket listeners receive datagrams addressed to the socket.) Then the sockets are opened to listen to any IP address. If the outgoing gateway has the **bind** command enabled and has an active call, the call becomes a one-way call with media flowing from the outgoing gateway to the terminating gateway. The dial peer bind socket listeners of the interface are closed and the configuration turns inactive for all subsequent SIP messages.
Not shut down Without active calls	TCP, TLS, and UDP socket listeners are initially closed. (Socket listeners receive datagrams addressed to the socket.) Then the sockets are opened and bound to the IP address set by the **bind** command. The sockets accept packets destined for the bound address only. The dial peer bind socket listeners of the interface are reopened and the configuration turns active for all subsequent SIP messages.
Not shut down With active calls	TCP, TLS, and UDP socket listeners are initially closed. Then the sockets are opened to listen to any IP address. The dial peer bind socket listeners of the interface are reopened and the configuration turns active for all subsequent SIP messages.

Interface State	Result Using bind all or bind control Commands
Bound interface's IP address is removed.	TCP, TLS, and UDP socket listeners are initially closed. Then the sockets are opened to listen to any address because the IP address has been removed. A message is printed that states the IP address has been deleted from the bound SIP interface. If the outgoing gateway has the **bind** command enabled and has an active call, the call becomes a one-way call with media flowing from the outgoing gateway to the terminating gateway. The dial peer bind socket listeners of the interface are closed and the configuration turns inactive for all subsequent SIP messages.
The physical cable is pulled on the bound port, or the interface layer goes down.	TCP, TLS, and UDP socket listeners are initially closed. Then the sockets are opened and bound to listen to any address. When the pulled cable is replaced, the result is as documented for interfaces that are not shut down. The dial peer bind socket listeners of the interface are closed and the configuration turns inactive for all subsequent SIP messages.
A bind interface is shut down, or its IP address is changed, or the physical cable is pulled while SIP calls are active.	The call becomes a one-way call with media flowing in only one direction. The media flows from the gateway where the change or shutdown took place to the gateway where no change occurred. Thus, the gateway with the status change no longer receives media. The call is then disconnected, but the disconnected message is not understood by the gateway with the status change, and the call is still assumed to be active. If the bind interface is shutdown, the dial peer bind socket listeners of the interface are closed. If the IP address of the interface is changed, the socket listeners representing the bind command is opened with the available IP address of the interface and the configuration turns active for all subsequent SIP messages.

How to Configure SIP Bind Features

Setting the Bind Command at the Global Level

To configure the **bind** command to an interface at the global level, perform the following steps.

Note The **bind media** command applies to specific interfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type* / *number*
4. **ip address** *ip-address mask* [**secondary**]
5. **exit**
6. **voice service voip**
7. **sip**
8. **bind** {**control** | **media** | **all**} **source-interface** *interface-id*[**ipv6-address** *ipv6-address*]
9. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type* / *number* **Example:** `Router(config)# interface fastethernet0/0`	Configures an interface type and enters the interface configuration mode. • *type* / *number* --Type of interface to be configured and the port, connector, or interface card number.
Step 4	**ip address** *ip-address mask* [**secondary**]	Configures a primary or secondary IP address for an interface.

	Command or Action	Purpose
	Example: `Router(config-if)# ip address 192.168.200.33 255.255.255.0`	• *ip-address mask* --IP address and mask for the associated IP subnet. • **secondary** --Makes the configured address a secondary IP address. If this keyword is omitted, the configured address is the primary IP address.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits the current mode.
Step 6	**voice service voip** **Example:** `Router(config)# voice service voip`	Enters voice service configuration mode.
Step 7	**sip** **Example:** `Router(conf-voi-serv)# sip`	Enters SIP configuration mode.
Step 8	**bind** {**control** | **media** | **all**} **source-interface** *interface-id*[**ipv6-address** *ipv6-address*] **Example:** `Router(conf-serv-sip)# bind control source-interface FastEthernet0/0`	Sets a source interface for signaling and media packets. • **control** --Binds signaling packets. • **media** --Binds media packets. • **all** --Binds signaling and media packets. • **source interface** *interface-id* --Type of interface and its ID. • **ipv6-address** *ipv6-address* --Configures the IPv6 address.
Step 9	**exit** **Example:** `Router(conf-serv-sip)# exit`	Exits the current mode.

Setting the Bind Command at the Dial-peer Level

To configure the **bind** command on SIP for a VoIP dial-peer, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type* / *number*
4. **ip address** *ip-address mask* [**secondary**]
5. **exit**
6. **dial-peer voice** *tag* **voip**
7. **session protocol sipv2**
8. **voice-class sip bind** {**control** | **media**} **source interface** *interface-id*[**ipv6-address** *ipv6-address*]
9. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type* / *number* **Example:** `Router(config)# interface fastethernet0/0`	Configures an interface type and enters the interface configuration mode. • *type* / *number* --Type of interface to be configured and the port, connector, or interface card number. **Note** You can only bind Loopback, Ethernet, FastEthernet, GigabitEthernet and Serial interfaces for dial peer.
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 2001:0DB8:0:1::1`	Configures a primary or secondary IP address for an interface. • *ip-address mask* --IP address and mask for the associated IP subnet. • **secondary** --Makes the configured address a secondary IP address. If this keyword is omitted, the configured address is the primary IP address.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits the current mode.

	Command or Action	Purpose
Step 6	**dial-peer voice** *tag* **voip** **Example:** `Router(config)# dial-peer voice 100 voip`	Enters dial peer voice configuration mode for the specified VoIP dial peer.
Step 7	**session protocol sipv2** **Example:** `Router(config-dial-peer)# session protocol sipv2`	Specifies use of IETF SIP.
Step 8	**voice-class sip bind** {**control** | **media**} **source interface** *interface-id*[**ipv6-address** *ipv6-address*] **Example:** `Router(config-dial-peer)# voice-class sip bind control source-interface fastethernet0/0 ipv6-address 2001:0DB8:0:1::1`	Sets a source interface for signaling and media packets. • **control** --Binds signaling packets. • **media** --Binds media packets. • **source interface** *interface-id* --Type of interface and its ID. • **ipv6-address** *ipv6-address* --(Optional) Configures the IPv6 address to the source interface.
Step 9	**exit** **Example:** `Router(config-dial-peer)# exit`	Exits the current mode.

Troubleshooting Tips

For troubleshooting tips and a list of important debug commands, see "Verifying and Troubleshooting SIP Features".

Monitoring the Bind Command

To monitor the **bind** command, perform the following steps.

SUMMARY STEPS

1. **show ip sockets**
2. **show sip-ua status**
3. **show sip-ua connections** {**tcp** [**tls**] | **udp**} {**brief** | **detail**}
4. **show dial-peer voice**

DETAILED STEPS

Step 1 **show ip sockets**

Use this command to display IP socket information and indicate whether the bind address of the receiving gateway is set.

The following sample output indicates that the bind address of the receiving gateway is set:

Example:

```
Router# show ip sockets
```

Proto Remote Port Local Port In Out Stat TTY OutputIF

17 0.0.0.0 0 --any-- 2517 0 0 9 0

17 --listen-- 172.18.192.204 1698 0 0 1 0

17 0.0.0.0 0 172.18.192.204 67 0 0 489 0

17 0.0.0.0 0 172.18.192.204 5060 0 0 A1 0

Example:

Step 2 **show sip-ua status**

Use this command to display SIP user-agent status and indicate whether bind is enabled.

The following sample output indicates that signaling is disabled and media on 172.18.192.204 is enabled:

Example:

```
Router# show sip-ua status
SIP User Agent Status
SIP User Agent for UDP : ENABLED
SIP User Agent for TCP : ENABLED
SIP User Agent for TLS over TCP : ENABLED
SIP User Agent bind status(signaling): DISABLED
SIP User Agent bind status(media): ENABLED 172.18.192.204
SIP early-media for 180 responses with SDP: ENABLED
SIP max-forwards : 70
SIP DNS SRV version: 2 (rfc 2782)
NAT Settings for the SIP-UA
Role in SDP: NONE
Check media source packets: DISABLED
Maximum duration for a telephone-event in NOTIFYs: 2000 ms
SIP support for ISDN SUSPEND/RESUME: ENABLED
Redirection (3xx) message handling: ENABLED
Reason Header will override Response/Request Codes: DISABLED
Out-of-dialog Refer: DISABLED
Presence support is DISABLED
protocol mode is ipv4
SDP application configuration:
 Version line (v=) required
Owner line (o=) required
 Timespec line (t=) required
 Media supported: audio video image
 Network types supported: IN
 Address types supported: IP4 IP6
 Transport types supported: RTP/AVP udptl
```

Step 3 **show sip-ua connections** {**tcp** [**tls**] | **udp**} {**brief** | **detail**}

Use this command to display the connection details for the UDP transport protocol. The command output looks identical for TCP and TLS.

Example:

```
Router# show sip-ua connections udp detail

Total active connections      : 0
No. of send failures          : 0
No. of remote closures        : 0
No. of conn. failures         : 0
No. of inactive conn. ageouts : 10
---------Printing Detailed Connection Report---------
Note:
 ** Tuples with no matching socket entry
    - Do 'clear sip <tcp[tls]/udp> conn t ipv4:<addr>:<port>'
      to overcome this error condition
 ++ Tuples with mismatched address/port entry
    - Do 'clear sip <tcp[tls]/udp> conn t ipv4:<addr>:<port> id <connid>'
      to overcome this error condition
 No Active Connections Found
-------------- SIP Transport Layer Listen Sockets ---------------
  Conn-Id               Local-Address
 ===========    ============================
   2            [9.42.28.29]:5060
```

Step 4 **show dial-peer voice**

Use this command, for each dial peer configured, to verify that the dial-peer configuration is correct. The following is sample output from this command for a VoIP dial peer:

Example:

```
Router# show dial-peer voice 101
VoiceOverIpPeer1234
        peer type = voice, system default peer = FALSE, information type = voice,
        description = `',
        tag = 1234, destination-pattern = `',
        voice reg type = 0, corresponding tag = 0,
        allow watch = FALSE
        answer-address = `', preference=0,
        CLID Restriction = None
        CLID Network Number = `'
        CLID Second Number sent
        CLID Override RDNIS = disabled,
        rtp-ssrc mux = system
        source carrier-id = `', target carrier-id = `',
        source trunk-group-label = `',  target trunk-group-label = `',
        numbering Type = `unknown'
        group = 1234, Admin state is up, Operation state is down,
        incoming called-number = `', connections/maximum = 0/unlimited,
        DTMF Relay = disabled,
        modem transport = system,
        URI classes:
            Incoming (Request) =
            Incoming (Via) =
            Incoming (To) =
            Incoming (From) =
            Destination =
        huntstop = disabled,
        in bound application associated: 'DEFAULT'
        out bound application associated: ''
        dnis-map =
        permission :both
        incoming COR list:maximum capability
        outgoing COR list:minimum requirement
        outgoing LPCOR:
```

```
Translation profile (Incoming):
Translation profile (Outgoing):
incoming call blocking:
translation-profile = `'
disconnect-cause = `no-service'
advertise 0x40 capacity_update_timer 25 addrFamily 4 oldAddrFamily 4
mailbox selection policy: none
type = voip, session-target = `',
technology prefix:
settle-call = disabled
ip media DSCP = ef, ip media rsvp-pass DSCP = ef
ip media rsvp-fail DSCP = ef, ip signaling DSCP = af31,
ip video rsvp-none DSCP = af41,ip video rsvp-pass DSCP = af41
ip video rsvp-fail DSCP = af41,
ip defending Priority = 0, ip preemption priority = 0
ip policy locator voice:
ip policy locator video:
UDP checksum = disabled,
session-protocol = sipv2, session-transport = system,
req-qos = best-effort, acc-qos = best-effort,
req-qos video = best-effort, acc-qos video = best-effort,
req-qos audio def bandwidth = 64, req-qos audio max bandwidth = 0,
req-qos video def bandwidth = 384, req-qos video max bandwidth = 0,
RTP dynamic payload type values: NTE = 101
Cisco: NSE=100, fax=96, fax-ack=97, dtmf=121, fax-relay=122
       CAS=123, TTY=119, ClearChan=125, PCM switch over u-law=0,
       A-law=8, GSMAMR-NB=117 iLBC=116, AAC-ld=114, iSAC=124
       lmr_tone=0, nte_tone=0
       h263+=118, h264=119
       G726r16 using static payload
       G726r24 using static payload
RTP comfort noise payload type = 19
fax rate = voice,   payload size =  20 bytes
fax protocol = system
fax-relay ecm enable
Fax Relay ans enabled
Fax Relay SG3-to-G3 Enabled (by system configuration)
fax NSF = 0xAD0051 (default)
codec = g729r8,   payload size =  20 bytes,
video codec = None
voice class codec = `'
voice class sip session refresh system
voice class sip rsvp-fail-policy voice post-alert mandatory keep-alive interval 30
voice class sip rsvp-fail-policy voice post-alert optional keep-alive interval 30
voice class sip rsvp-fail-policy video post-alert mandatory keep-alive interval 30
voice class sip rsvp-fail-policy video post-alert optional keep-alive interval 30
text relay = disabled
Media Setting = forking (disabled) flow-through (global)
Expect factor = 10, Icpif = 20,
Playout Mode is set to adaptive,
Initial 60 ms, Max 1000 ms
Playout-delay Minimum mode is set to default, value 40 ms
Fax nominal 300 ms
Max Redirects = 1, signaling-type = cas,
VAD = enabled, Poor QOV Trap = disabled,
Source Interface = NONE
voice class sip url = system,
voice class sip tel-config url = system,
voice class sip rel1xx = system,
voice class sip anat = system,
voice class sip outbound-proxy = "system",
voice class sip associate registered-number =
                 system,
voice class sip asserted-id system,
voice class sip privacy system
voice class sip e911 = system,
voice class sip history-info = system,
voice class sip reset timer expires 183 = system,
voice class sip pass-thru headers = system,
voice class sip pass-thru content unsupp = system,
voice class sip pass-thru content sdp = system,
```

```
voice class sip copy-list = system,
voice class sip g729 annexb-all = system,
voice class sip early-offer forced = system,
voice class sip negotiate cisco = system,
voice class sip block 180 = system,
voice class sip block 183 = system,
voice class sip block 181 = system,
voice class sip preloaded-route = system,
voice class sip random-contact = system,
voice class sip random-request-uri validate = system,
voice class sip call-route p-called-party-id = system,
voice class sip call-route history-info = system,
voice class sip privacy-policy send-always = system,
voice class sip privacy-policy passthru = system,
voice class sip privacy-policy strip history-info = system,
voice class sip privacy-policy strip diversion = system,
voice class sip map resp-code 181 = system,
voice class sip bind control = enabled, 9.42.28.29,
voice class sip bind media = enabled, 9.42.28.29,
voice class sip bandwidth audio = system,
voice class sip bandwidth video = system,
voice class sip encap clear-channel = system,
voice class sip error-code-override options-keepalive failure = system,
voice class sip calltype-video = false
voice class sip registration passthrough = System
voice class sip authenticate redirecting-number  = system,
redirect ip2ip = disabled
local peer = false
probe disabled,
Secure RTP: system (use the global setting)
voice class perm tag = `'
Time elapsed since last clearing of voice call statistics never
Connect Time = 0, Charged Units = 0,
Successful Calls = 0, Failed Calls = 0, Incomplete Calls = 0
Accepted Calls = 0, Refused Calls = 0,
Last Disconnect Cause is "",
Last Disconnect Text is "",
Last Setup Time = 0.
Last Disconnect Time = 0.
```

Note If the bind address is not configured at the dial-peer, the output of the **show dial-peer voice** command remains the same except for the values of the **voice class sip bind control** and **voice class sip bind media**, which display “system”, indicating that the bind is configured at the global level.

Troubleshooting Tips

For troubleshooting tips and a list of important debug commands, see "Verifying and Troubleshooting SIP Features".

Configuration Examples for SIP Bind Features

Example Verifying the bind Command

This sample output shows that bind is enabled on router 172.18.192.204:

```
Router# show running-config
Building configuration...
```

```
Current configuration : 2791 bytes
!
version 12.2
service config
no service single-slot-reload-enable
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service internal
service udp-small-servers
!
ip subnet-zero
ip ftp source-interface Ethernet0
!
voice service voip
 sip
  bind control source-interface FastEthernet0
!
interface FastEthernet0
 ip address 172.18.192.204 255.255.255.0
 duplex auto
 speed auto
 fair-queue 64 256 1000
 ip rsvp bandwidth 75000 100
!
voice-port 1/1/1
no supervisory disconnect lcfo
!
dial-peer voice 1 pots
application session
destination-pattern 5550111
port 1/1/1
!
dial-peer voice 29 voip
application session
destination-pattern 5550133
session protocol sipv2
session target ipv4:172.18.200.33
codec g711ulaw
!
gateway
!
line con 0
line aux 0
line vty 0 4
login
!
end
```

Additional References

Related Documents

Related Topic	Document Title
SIP Overview	"Overview of SIP"
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Voice commands	*Cisco IOS Voice Command Reference*

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
CISCO-SIP-UA-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2543	SIP: Session Initiation Protocol
RFC 2806	URLs for Telephone Calls

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

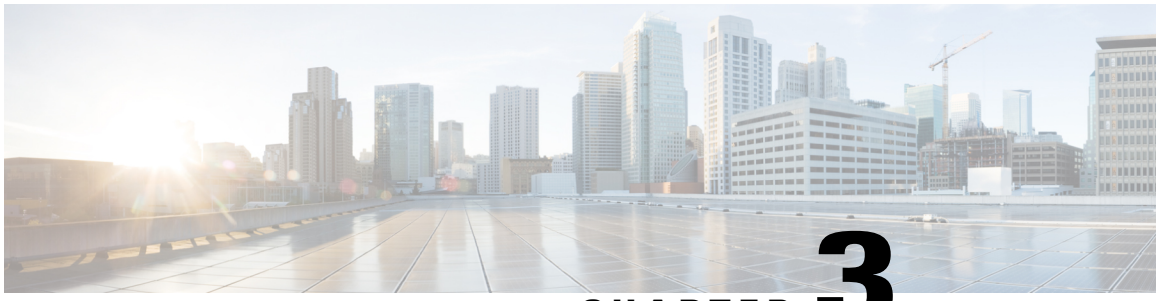
Feature Information for SIP Bind Features

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 5: Feature Information for SIP Bind Features

Feature Name	Releases	Feature Information
SIP Gateway Support for the bind Command	12.2(2)XB 12.2(2)XB2 12.2(8)T 12.2(11)T 12.3(4)T Cisco IOS XE Release 3.1.0S	The SIP Gateway Support for the **bind** command feature allows you to configure the source IP address of signaling packets and media packets. In 12.2(2)XB, this feature was introduced. In 12.3(4)T, this feature was expanded to provide the flexibility to specify different source interfaces for signaling and media, and allow network administrators a finer granularity of control on the network interfaces used for voice traffic. The following commands were introduced or modified: **bind**, **show dial-peer voice**, **show ip sockets**, **show sip-ua connections**, and **show sip-ua status**.
Support Ability to Configure Source IP Address for Signaling and Media per SIP Trunk	15.1(2)T	This feature allows you to configure a separate source IP address per SIP trunk. This source IP address is embedded in all SIP signaling and media packets that traverse the SIP trunk. This feature enables service providers for better profiling and billing policies. It also enables greater security for enterprises by the use of distinct IP addresses within and outside the enterprise domain. The following command was introduced or modified: **voice-class sip bind**.



CHAPTER 3

Configuring Media Path

The Media Path feature allows you to configure the path taken by media after a call is established. You can configure Media Path in the following modes:

- Media Flow-Through, on page 34
- Media Flow-Around, on page 35
- Media Anti-Trombone, on page 35

For information on Delayed Offer to Early Offer, refer to Delayed-Offer to Early-Offer

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for Configuring Media Path

For Media Flow-Through

- Video codecs are not supported for Media Flow-Through.
- Media flow-around for Delayed-Offer to Early-Offer audio and video calls is not supported.

For Media Anti-Tromboning

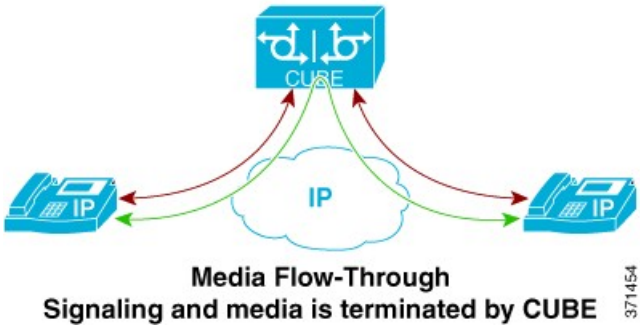
- When Media Anti-Tromboning media path mode is activated, Cisco UBE does not perform supplementary services such as handling REFER-based call transfers or media services such as Secure Real-Time Transport Protocol (SRTP) and SNR.
- Anti-Tromboning does not work if one call leg is media flow-through and the other call leg is Media Flow-Around. Similarly, Anti-Tromboning does not work if one call leg is Session Description Protocol (SDP) pass-through and another call leg is SDP normal.
- H.323 is not supported.

Information About Media Path

Media Flow-Through

Media Flow-Through is a media path mode where media and signaling packets terminate and originate on CUBE. As CUBE is an active participant of the call, this mode is recommended when connected outside an enterprise (untrusted endpoints).

Figure 8: Media Flow-Through Mode

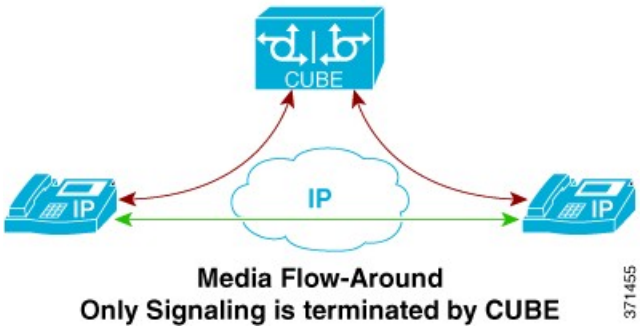


The **media flow-through** command is used to configure this feature in global VoIP configuration mode (config-voi-serv), dial-peer configuration mode (config-dial-peer) and voice-class configuration mode (config-class). Refer to Modes for Configuring Dial Peers to enter these modes and configure this feature.

Media Flow-Around

Media Flow-Around is a media path mode where signaling packets terminate and originate on CUBE. As media bypasses CUBE and flows directly between endpoints, this mode is recommended when connected within an enterprise (trusted endpoints). Media Flow-Around is supported for both audio and video calls.

Figure 9: Media Flow-Around



The **media flow-around** command is used to configure this feature in global VoIP configuration mode (config-voi-serv), dial peer mode (config-dial-peer) and voice class configuration mode (config-class). Refer to Modes for Configuring Dial Peers to enter these modes and configure this feature.

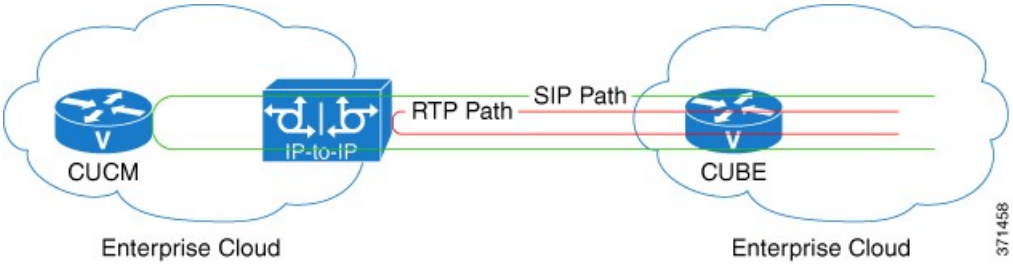
Media Anti-Trombone

Media Anti-Tromboning is a media path mode that allows CUBE to detect and avoid loops created by call transfers or call forwards. Loops are restricted to the SIP signaling path and removed from the RTP media path.

The user agent may initiate call forwards and call transfers that are sent towards CUBE as a new SIP INVITE dialog. CUBE considers the original call and the forwarded call as separate unrelated calls. Media anti-tromboning allows CUBE to detect the relation between the calls and resolve the media loop by sending SDP packets back to the sender.

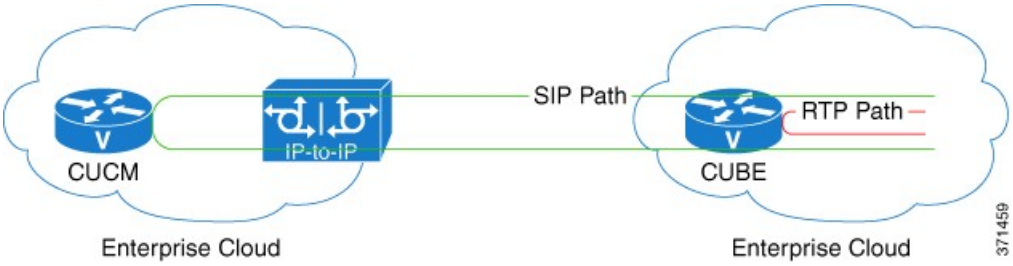
The figure below illustrates how CUBE needlessly loops RTP packets towards the User Agent because it fails to detect the loop.

Figure 10: Tromboning - Needless looping of Media Packets



The figure below illustrates how CUBE detects and avoids the loop with the anti-tromboning feature.

Figure 11: Anti-Tromboning - Avoiding Media Loops



The **media anti-trombone** command is used to configure this feature in global VoIP configuration mode (config-voi-serv), dial-peer configuration mode (config-dial-peer) and voice-class configuration mode (config-class). Refer to Modes for Configuring Dial Peers to enter these modes and configure this feature.

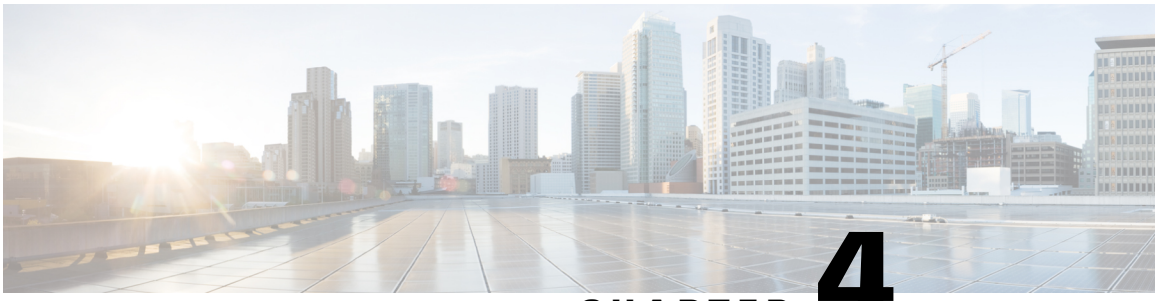
Feature Information for Media Path

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 6: Feature Information for Configuring Path of Media

Feature Name	Releases	Feature Information
Configuring Media Path	12.4(3), 12.4(24)T, 15.0(1)M	The Media Path feature allows you to configure the path taken by media after a call is established. The following commands were introduced by this feature: **media-flow around, media flow-through, media anti-trombone**.



CHAPTER 4

SIP Profiles

Session Initiation Protocol (SIP) profiles change SIP incoming or outgoing messages so that interoperability between incompatible devices can be ensured.

SIP profiles can be configured with rules to add, remove, copy, or modify the SIP, Session Description Protocol (SDP), and peer headers that enter or leave CUBE.

Figure 12: Incoming and Outgoing messages where SIP Profiles can be applied

You can use the following tool to test your SIP profile on an incoming message.
http://cantor.cisco.com/sip-profiles.html

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for SIP Profiles

- Removal or addition of mandatory headers is not supported. You can only modify mandatory headers Mandatory SIP headers include To, From, Via, CSeq, Call-Id, and Max-Forwards. Mandatory SDP headers include v, o, s, t ,c, and m.
- Addition or removal of entire Multipurpose Internet Mail Extensions (MIME) or (Session Description Protocol) SDP bodies from SIP messages.
- Syntax checking is not performed on SIP messages after SIP profile rules have been applied. Changes specified in the SIP profile should result in valid SIP protocol exchanges.
- The header length (including header name) after modification should not exceed 300 characters. Max header length for add value is approximately 220 characters. Max SDP length is 2048 characters. If any header length exceeds this maximum value after applying SIP profiles, then the profile is not applied.
- If a header-name is changed to its compact form, SIP profile rules cannot be applied on that header. Thus a SIP profile rule modifying a header name to its compact form must be the last rule on that header.
- We cannot modify the "image" m-line attributes (m=image 16850 udptl t38) using SIP profiles. SIP profiles can be applied only on audio and video m-lines in SDP.
- In a high-availability (HA) scenario, SIP profiles copy variable data is not check-pointed to standby.
- Existing limitations and restrictions of outbound SIP profiles apply to inbound SIP profiles as well.

SIP Copylist:

- You cannot configure more than 99 variables for the SIP profiles copy option.
- This feature does not support any header other than SIP.

Information About SIP Profile

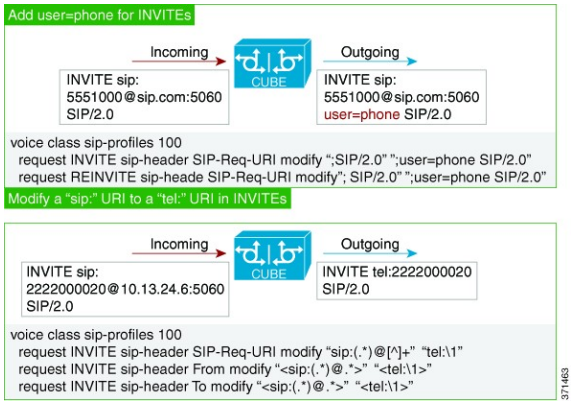
SIP Profile

Protocol translation and repair is a key Cisco Unified Border Element (CUBE) function. CUBE can be deployed between two devices that support the same VoIP protocol (For example. SIP), but do not interwork because of differences in how the protocol is implemented or interpreted. CUBE can customize the SIP messaging on either side to what the devices in that segment of the network expects to see by normalizing the SIP messaging on the network border, or between two non-interoperable devices within the network.

Service providers may have policies for which SIP messaging fields should be present (or what constitutes valid values for the header fields) before a SIP call enters their network. Similarly, enterprises and small

businesses may have policies for the information that can enter or exit their networks for policy or security reasons from a service provider SIP trunk.

Figure 13: SIP Profile



In order to customize SIP messaging in both directions, you can place and configure a CUBE with a SIP profile at the boundary of these networks.

In addition to network policy compliance, the CUBE SIP profiles can be used to resolve incompatibilities between SIP devices inside the enterprise network. These are the situations in which incompatibilities can arise:

- A device rejects an unknown header (value or parameter) instead of ignoring it
- A device sends incorrect data in a SIP message
- A device does not implement (or implements incorrectly) protocol procedures
- A device expects an optional header value or parameter, or an optional protocol procedure that can be implemented in multiple ways
- A device sends a value or parameter that must be changed or suppressed before it leaves or enters the network
- Variations in the SIP standards on how to achieve certain functions

The SIP profiles feature on CUBE provides a solution to these incompatibilities and customization issues.

SIP profiles can also be used to change a header name from the long form to the compact form. For example, From to f. This can be used as a way to reduce the length of a SIP message. By default, the device never sends the compact form of the SIP messages although it receives either the long or the short form.

Important Notes for SIP Profiles

Given below are a few important notes for SIP Profiles:

- Copy Variables u01 to u99 are shared by inbound and outbound SIP Profiles.
- Session Initiation Protocol (SIP) and Session Description Protocol (SDP) headers are supported. SDP can be either a standalone body or part of a Multipurpose Internet Mail Extensions (MIME) message.

- The rules configured for an INVITE message are applied only to the first INVITE of a call. A special REINVITE keyword is used to manipulate subsequent INVITEs of a CALL.
- Manipulation of SIP headers by outbound SIP profiles occurs as the last step before the message leaves the CUBE device; that is, after destination dial-peer matching has taken place. Changes to the SIP messages are not remembered or acted on by the CUBE application. The Content-length field is recalculated after the SIP Profiles rules are applied to the outgoing message.
- The **ANY** keyword indicates that a rule must be applied to any message within the specified category.
- SIP header modification can be cryptic. It is easier to remove a header and add it back (with the new value), rather than modifying it.
- To include '?' (question-mark) character as part of match-pattern or replace-pattern, you need to press "Ctrl+v" keys and then type '?'. This is needed to treat '?' as a input character itself instead of usual device help prompt.
- For header values used to add, modify or copy a header:
  - If a whitespace occurs, the entire value must be included between double quotes. For example, "User-Agent: CISCO CUBE"
  - If double quotes occurs, a back slash must prefix the double quotes. For example, "User-Agent: \"CISCO\" CUBE"
  - Regular expressions are supported.

Inbound SIP Profile:

- If the incoming message contains multiple instances of same header, the header values are stored as a comma separated list, and this needs to be considered while modifying it.
- Modification by an inbound SIP profile takes place before regular SIP call processing happens so that behavior of CUBE would be as if it received the message directly without modification.

  If inbound dial peer matching fails as required information could not be extracted from headers (like Request-URI, Via, From or To) due to issues in them, global dial peers are applied. An example is a request with invalid SIP-Req-URI.
- After modification by inbound SIP Profiles, the parameters in SIP message might change, which might change the inbound dial-peer matched when actual dial-peer lookup is done.
- In the register pass-through feature, there is only one dial-peer for register and response. So both register from phone and response from registrar would go through the same inbound sip profile under the dial-peer if any.

SIP Copylist - Passing Unsupported Parameters of a Mandatory Header

A SIP copylist is used to pass contents of headers in an incoming dial peer to an outgoing dial peer. This feature is used to pass unsupported, parameters of a mandatory headers from one leg to another. When a SIP message is received, a check is done for the header, and if it is available, it is copied into a list and passed on to the outbound dial peer leg.

Copying Content From a Header to a Header of an Outgoing Message

Using a SIP profile with a **copy** and a **modify** rule configured:

- You can copy content from the header of an incoming message (peer header) to the header of an outgoing message
- You can copy content from the header of one outgoing message to the header of another outgoing message

Content from headers are copied into copy variables in the copy rule and pasted into other headers in the modify rule. If a header in a mandatory message is not supported by CUBE, configure passing of that header using a copylist and apply the rule to the incoming message.

How to Configure SIP Profiles

To configure SIP Profiles, you must first configure the SIP Profile globally, and apply it at either to all dial peers (globally) or to a single dial peer (dial-peer level). Once a SIP profile is configured, it can be applied as an inbound or outbound profile.

Configuring a SIP Profile to Manipulate SIP Request Headers

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice class sip-profiles** *profile-id*
4. Add, remove, modify, or copy SIP headers or responses:
   - **request** *message* {**sip-header** | **sdp-header**} *header-to-add* **add** *header-value-to-add*
   - **request** *message* {**sip-header** | **sdp-header**} *header-to-remove* **remove**
   - **request***message* {**sip-header** | **sdp-header**} *header-to-modify* **modify** *header-value-to-match header-value-to-replace*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal**	Enters global configuration mode.
Step 3	**voice class sip-profiles** *profile-id* **Example:** `Device(config)# voice class sip-profiles 10`	Creates a SIP Profiles and enters voice class configuration mode.
Step 4	Add, remove, modify, or copy SIP headers or responses: • **request** *message* {**sip-header** | **sdp-header**} *header-to-add* **add** *header-value-to-add* • **request** *message* {**sip-header** | **sdp-header**} *header-to-remove* **remove** • **request***message* {**sip-header** | **sdp-header**} *header-to-modify* **modify** *header-value-to-match header-value-to-replace*	According to your choice, this step does one of the following: • Adds a SIP or SDP header to a SIP request. • Removes a SIP or SDP header to a SIP request. • Modifies a SIP or SDP header to a SIP request. • The **ANY** keyword indicates that a rule must be applied to any message within the specified category. • For *header-value-to-add* used to add a header, *header-value-to-match* or *header-value-to-replace* used to modify a header: ◦ If a whitespace occurs, the entire value must be included between double quotes. For example, “User-Agent: CISCO CUBE” ◦ If double quotes occurs, a back slash must prefix the double quotes. For example, “User-Agent: \”CISCO\” CUBE” ◦ Regular expressions are supported. Refer to Example: Adding a SIP, SDP, or Peer Header, on page 60, Example: Modifying a SIP, SDP, or Peer Header, on page 61, and Example: Remove a SIP, SDP, or Peer Header, on page 63 for more details.
Step 5	**end**	Exits to privileged EXEC mode

What to Do Next

Now apply the SIP Profile as an inbound or outbound SIP profile.

Configuring a SIP Profile to Manipulate SIP Response Headers

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice class sip-profiles** *profile-id*
4. Add, remove, modify, or copy SIP response headers:
   - **response** *message* [**method** *method-type*] {**sip-header** | **sdp-header**} *header-to-add* **add** *header-value-to-add*
   - **response** *message* [**method** *method-type*] {**sip-header** | **sdp-header**} *header-to-remove* **remove**
   - **response** *message* [**method** *method-type*] {**sip-header** | **sdp-header**} *header-to-modify* **modify** *header-value-to-match header-value-to-replace*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal**	Enters global configuration mode.
Step 3	**voice class sip-profiles** *profile-id* **Example:** `Device(config)# voice class sip-profiles 10`	Creates a SIP Profiles and enters voice class configuration mode.
Step 4	Add, remove, modify, or copy SIP response headers: • **response** *message* [**method** *method-type*] {**sip-header**	**sdp-header**} *header-to-add* **add** *header-value-to-add* • **response** *message* [**method** *method-type*] {**sip-header**	**sdp-header**} *header-to-remove* **remove** • **response** *message* [**method** *method-type*] {**sip-header**		According to your choice, this step does one of the following: • Adds a SIP or SDP header to a SIP response. • Removes a SIP or SDP header to a SIP response. • Modifies a SIP or SDP header to a SIP response. • The **ANY** keyword indicates that a rule must be applied to any message within the specified category. • You can modify a SIP response status line, using the **SIP-StatusLine** SIP header of the response. • The **method** keyword is used to associate a response to its corresponding request before application of the SIP profile manipulations. For example,

	Command or Action	Purpose
	sdp-header} *header-to-modify* **modify** *header-value-to-match header-value-to-replace*	a SIP profile configured without the **method** keyword for a 200 response code is applied to the 200 response code for all requests such as INVITE, UPDATE, BYE, PRACK. But the **method** keword allows you to selectively apply the profiles based on the request to which the response is sent. • For *header-value-to-add* used to add a header, *header-value-to-match* or *header-value-to-replace* used to modify a header: ◦ If a whitespace occurs, the entire value must be included between double quotes. For example, “User-Agent: CISCO CUBE” ◦ If double quotes occurs, a back slash must prefix the double quotes. For example, “User-Agent: \”CISCO\” CUBE” ◦ Regular expressions are supported. Refer to Example: Adding a SIP, SDP, or Peer Header, on page 60, Example: Modifying a SIP, SDP, or Peer Header, on page 61, and Example: Remove a SIP, SDP, or Peer Header, on page 63 for more details.
Step 5	**end**	Exits to privileged EXEC mode

What to Do Next

Now apply the SIP Profile as an inbound or outbound SIP profile.

Configuring a SIP Profile as an Outbound Profile

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Apply the SIP profile to a dial peer:
   - **voice-class sip profiles** *profile-id* in the dial-peer configuration mode.
   - **sip-profiles** *profile-id* in the global VoIP configuration mode
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal**	Enters global configuration mode.

Step 3	Apply the SIP profile to a dial peer:
	• **voice-class sip profiles** *profile-id* in the dial-peer configuration mode.
	• **sip-profiles** *profile-id* in the global VoIP configuration mode
	Example:
	In dial-peer configuration mode
	`!Applying SIP profiles to one dial peer only`
	`Device (config) dial-peer voice 10 voip`
	`Device (config-dial-peer) voice-class sip profiles 30`
	`Device (config-dial-peer) end`
	Example:
	In global VoIP SIP mode
	`! Applying SIP profiles globally`
	`Device(config)# voice service voip`
	`Device (config-voi-serv) sip`
	`Device (config-voi-sip) sip-profiles 20`
	`Device (config-voi-sip) end`	

Step 4	**end**	Exits to privileged EXEC mode .

Configuring a SIP Profile as an Inbound Profile

You can configure a SIP profile as an inbound profile applied globally or to a single inbound dial peer. Inbound SIP profiles feature must be enabled before applying it to dial peers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **sip**
5. **sip-profiles inbound**
6. Apply the SIP profile to a dial peer:
   - **voice-class sip profiles** *profile-id* **inbound** in the dial-peer configuration mode.
   - **sip-profiles** *profile-id* **inbound** in the global VoIP configuration mode
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal**	Enters global configuration mode.
Step 3	**voice service voip** **Example:** `Device(config)# voice service voip`	Enters global VoIP configuration mode.
Step 4	**sip** **Example:** `Device(config-voi-serv)# sip`	Enters global VoIP SIP configuration mode.
Step 5	**sip-profiles inbound** **Example:** `Device(config-voi-sip)# sip-profiles inbound`	Enables inbound SIP profiles feature.
Step 6	Apply the SIP profile to a dial peer: • **voice-class sip profiles** *profile-id* **inbound** in the dial-peer configuration mode. • **sip-profiles** *profile-id* **inbound** in the global VoIP configuration mode	

	Command or Action	Purpose

	Example:
	In dial-peer configuration mode
	`!Applying SIP profiles to one dial peer only`
	`Device (config)# dial-peer voice 10 voip`
	`Device (config-dial-peer)# voice-class sip profiles 30 inbound`
	`Device (config-dial-peer)# end`
	Example:
	In global VoIP SIP mode
	`! Applying SIP profiles globally`
	`Device(config)# voice service voip`
	`Device (config-voi-serv)# sip`
	`Device (config-voi-sip)# sip-profiles 20 inbound`
	`Device (config-voi-sip)# end`	

Step 7	**end**	Exits to privileged EXEC mode

Verifying SIP Profiles

SUMMARY STEPS

1. **show dial-peer voice** *id* | **include profile**

DETAILED STEPS

show dial-peer voice *id* | **include profile**

Example:

```
Device# show dial-peer voice 10 | include profile

        Translation profile (Incoming):
        Translation profile (Outgoing):
        translation-profile = `'
        voice class sip profiles = 11
        voice class sip profiles inbound = 10
```

Displays information related to SIP profiles configured on the specified dial peer.

Troubleshooting SIP Profiles

SUMMARY STEPS

1. **debug ccsip all**

DETAILED STEPS

debug ccsip all
This command displays the applied SIP profiles.

Example:
Applied SIP profile is highlighted in the example below.

```
Device# debug ccsip all
...
Oct 12 06:51:53.619: //-1/735085DC8F3D/SIP/Info/sipSPIGetShrlPeer:
                     Try match incoming dialpeer for Calling number:
                     : sippOct 12 06:51:53.619:
                     //-1/735085DC8F3D/SIP/Info/sipSPIGetCallConfig:
                     Peer tag 2 matched for incoming call
Oct 12 06:51:53.619: //-1/xxxxxxxxxxxx/SIP/Info/sipSPIGetCallConfig:
                     voice class SIP profiles tag is set : 1
Oct 12 06:51:53.619: //-1/735085DC8F3D/SIP/Info/sipSPIGetCallConfig:
                     Not using Voice Class Codec
Oct 12 06:51:53.619: //-1/735085DC8F3D/SIP/Info/sipSPIGetCallConfig:
                     xcoder high-density disabled
Oct 12 06:51:53.619: //-1/735085DC8F3D/SIP/Info/sipSPIGetCallConfig:
                     Flow Mode set to FLOW_THROUGH
```

This command also displays the modifications performed by the SIP profile configuration, by preceding the modification information with the word sip_profiles, as highlighted in the example below.

Example:

```
Device# debug ccsip all
...
Oct 12 06:51:53.647: //-1/xxxxxxxxxxxx/SIP/Info/
                     sip_profiles_application_change_sdp_line:
                     New SDP header is added : b=AS: 1600
Oct 12 06:51:53.647: //-1/xxxxxxxxxxxx/SIP/Info/
                     sip_profiles_update_content_length:
                     Content length header before modification :
                     Content-Length: 290
Oct 12 06:51:53.647: //-1/xxxxxxxxxxxx/SIP/Info/
                     sip_profiles_update_content_length:
                     Content length header after modification :
                     Content-Length: 279
```

How to Copy Headers to Another Using SIP Profiles

Copying SIP headers from one message (request or response) to another is possible in one of the following ways:

- For an incoming SIP message, you can enable the copying of an unsupported mandatory header to the corresponding outbound call leg using a SIP copylist. This is done using the **sip-header SIP-Req-URI** or **sip-header SIP-Req-URI** command.
- You can copy content from the header of an incoming message (peer header) to the header of an outgoing message. The incoming message has to be enabled as described in the previous note and copied to a user-defined variable that can then be applied to the outgoing SIP header. This is done using a **copy** or **modify** keyword.
- You can copy content from the header of one outgoing message to the header of another outgoing message. This is done using a **copy** or **modify** keyword.

Copying Contents From an Incoming Header and Modifying the Outgoing Header

To copy content from an incoming header that a device receives to an outgoing header, configure a SIP copylist for that header and apply it to an incoming dial peer. A SIP profile is configured to copy this incoming header to a user-defined variable and apply it to an outgoing header.

Before You Begin

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice class sip-copylist** *tag*
4. Do one of the following:
   - **sip-header** *header-name*
   - **sip-header SIP-Req-URI**
5. **exit**
6. **dial-peer voice** *inbound-dial-peer-tag* **voip**
7. **voice class sip-copylist** *tag*
8. **exit**
9. **voice class sip-profiles** *profile-id*
10. {**request** | **response**} *message* **peer-header sip** *header-to-copy* **copy** *header-value-to-match copy-variable*
11. {**request** | **response**} *message* {**sip-header** | **sdp-header**} *header-to-modify* **modify** *header-value-to-match header-value-to-replace*
12. **exit**
13. **dial-peer voice** *inbound-dial-peer-tag* **voip**
14. **voice class sip-copylist** *tag*
15. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal**	Enters global configuration mode.
Step 3	**voice class sip-copylist** *tag* **Example:** `Device(config)# voice class sip-copylist 100`	Configures a list of entities to be sent to a peer call leg and enters voice class configuration mode.
Step 4	Do one of the following: • **sip-header** *header-name* • **sip-header SIP-Req-URI** **Example:** `Device(config-class)# sip-header To`	Specifies the SIP header to be copied to the peer call leg. • **sip-req-uri—**Configures Cisco Unified Border Element (UBE) to send a SIP request Uniform Resource Identifier (URI) to the peer call leg. • *header-name***—**Configures Cisco Unified Border Element (UBE) to send the header name specified to the peer call leg.
Step 5	**exit**	Exits voice class configuration mode.
Step 6	**dial-peer voice** *inbound-dial-peer-tag* **voip** **Example:** `Device(config)# dial-peer voice 2 voip`	Enters the dial peer configuration mode for the specified inbound dial peer.
Step 7	**voice class sip-copylist** *tag* **Example:** `Device(config-dial-peer)# voice class sip-copylist 100`	Applies the copy list to the dial-peer.
Step 8	**exit**	Exits to global configuration mode.
Step 9	**voice class sip-profiles** *profile-id* **Example:** `Device(config)# voice class sip-profiles 10`	Creates a SIP Profiles and enters voice class configuration mode.
Step 10	{**request** | **response**} *message* **peer-header sip** *header-to-copy* **copy** *header-value-to-match copy-variable*	Copies headers from the corresponding incoming dial peer into a copy variable.

	Command or Action	Purpose
	Example: `Device(config-class)# request INVITE peer-header sip TO copy "sip:(.*)@" u01`	
Step 11	{**request** | **response**} *message* {**sip-header** | **sdp-header**} *header-to-modify* **modify** *header-value-to-match header-value-to-replace* **Example:** `Device(config-class)# request INVITE sip-header SIP-Req-URI modify ".*@(.*)" "INVITE sip:\u01@\1"`	Modifies an outgoing SIP or SDP header using the copy variable defined in the previous step.
Step 12	**exit**	Exits to global configuration mode.
Step 13	**dial-peer voice** *inbound-dial-peer-tag* **voip** **Example:** `Device(config)# dial-peer voice 2 voip`	Enters the dial peer configuration mode for the specified inbound dial peer.
Step 14	**voice class sip-copylist** *tag* **Example:** `Device(config-dial-peer)# voice class sip-copylist 100`	Applies the copy list to the dial-peer.
Step 15	**exit**	Exits to global configuration mode.

What to Do Next

Apply the SIP profile to an outbound dial peer.

Copying Contents From an Outgoing Header and Modifying Another Outgoing Header

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice class sip-profiles** *profile-id*
4. {**request** | **response**} *message* {**sip-header** | **sdp-header**} *header-to-copy* **copy** *header-value-to-match copy-variable*
5. {**request** | **response**} *message* {**sip-header** | **sdp-header**} *header-to-modify* **modify** *header-value-to-match header-value-to-replace*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal**	Enters global configuration mode.
Step 3	**voice class sip-profiles** *profile-id* **Example:** `Device(config)# voice class sip-profiles 10`	Creates a SIP Profiles and enters voice class configuration mode.
Step 4	{**request** | **response**} *message* {**sip-header** | **sdp-header**} *header-to-copy* **copy** *header-value-to-match copy-variable* **Example:** `Device(config-class)# request INVITE sip-header TO copy "sip:(.*)@" u01`	Copies the contents of the specified header from an outbound message into a copy variable.
Step 5	{**request** | **response**} *message* {**sip-header** | **sdp-header**} *header-to-modify* **modify** *header-value-to-match header-value-to-replace* **Example:** `Device(config-class)# request INVITE sip-header SIP-Req-URI modify ".*@(.*)" "INVITE sip:\u01@\1"`	Modifies an outgoing SIP or SDP header using the copy variable defined in the previous step.
Step 6	**end** **Example:** `Device(config-class)# end`	Exits voice class configuration mode and enters privileged EXEC mode.

What to Do Next

Apply the SIP Profile to an outbound dial peer.

How to Manipulate the Status-Line Header of SIP Responses Using SIP Profiles

The SIP status line is a SIP response header, and it can be modified like any other SIP headers of a message. it can either be modified with a user-defined value, or the status line from an incoming response can be copied to an outgoing SIP response. The SIP header keyword used for the response status line is **SIP-StatusLine**.

You can copying the SIP response status-lines from one leg to another with the following steps

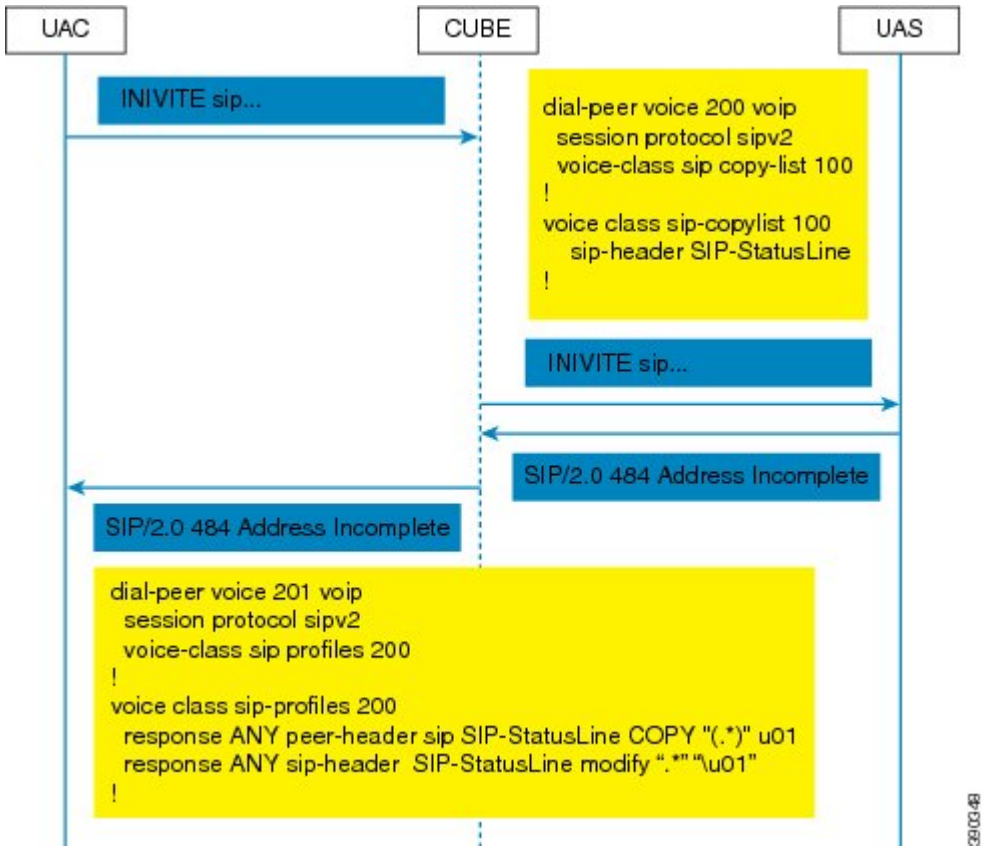
1. For an incoming SIP response, enable the copying of status line on the corresponding dial peer, by adding the status line to a copylist (list of headers to be copied) associated with a dial peer. This is done using the **sip-header SIP-StatusLine** command inside the copylist.
2. For an outgoing SIP response, enable the copying of the previously enabled SIP response to a user-defined variable that can then be applied to the outgoing SIP response. This is done using a conditional profile with a **sip-StatusLine copy** or **modify** keyword. See the call flow in the following figure:

Copying Incoming SIP Response Status Line to Outgoing SIP Response

To copy content from the status line of an incoming SIP response that a device receives to an outgoing response, configure a SIP copylist for SIP status line and apply it to an incoming dial peer. A SIP profile must be

configured to copy the status line of an incoming SIP response to a user-defined variable and apply it to an outgoing SIP response.

Figure 14: Call Flow for Copying the Status Line from the Incoming SIP Response to the Outgoing SIP Response



SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice class sip-copylist** *tag*
4. **sip-header SIP-StatusLine**
5. **exit**
6. **dial-peer voice** *inbound-dial-peer-id* **voip**
7. **voice-class sip copy-list** *list-id*
8. **exit**
9. **voice class sip-profiles** *tag*
10. **response** *response-code* **peer-header sip SIP-StatusLine copy** *match-pattern copy-variable*
11. **response** *response-code* **sip-header SIP-StatusLine modify** *match-pattern copy-variable*
12. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**voice class sip-copylist** *tag* **Example:** `Device(config)# voice class sip-copylist 1`	Configures a list of entities to be sent to the peer call leg and enters voice class configuration mode.
Step 4	**sip-header SIP-StatusLine** **Example:** `Device(config-class)# sip-header SIP-StatusLine`	Specifies that the Session Initiation Protocol (SIP) status line header must be sent to the peer call leg.
Step 5	**exit** **Example:** `Device(config-class)# exit`	Exits voice class configuration mode and returns to global configuration mode.
Step 6	**dial-peer voice** *inbound-dial-peer-id* **voip** **Example:** `Device(config)# dial-peer voice 99 voip`	Specifies an inbound dial peer and enters dial peer configuration mode.
Step 7	**voice-class sip copy-list** *list-id* **Example:** `Device(config-dial-peer)# voice-class sip copy-list 1`	Associates the SIP copy list with the inbound dial peer.
Step 8	**exit** **Example:** `Device(config-dial-peer)# exit`	Exits dial peer configuration mode and returns to global configuration mode.
Step 9	**voice class sip-profiles** *tag* **Example:** `Device(config)# voice class sip-profiles 10`	Enables dial peer-based VoIP SIP profile configurations and enters voice class configuration mode.

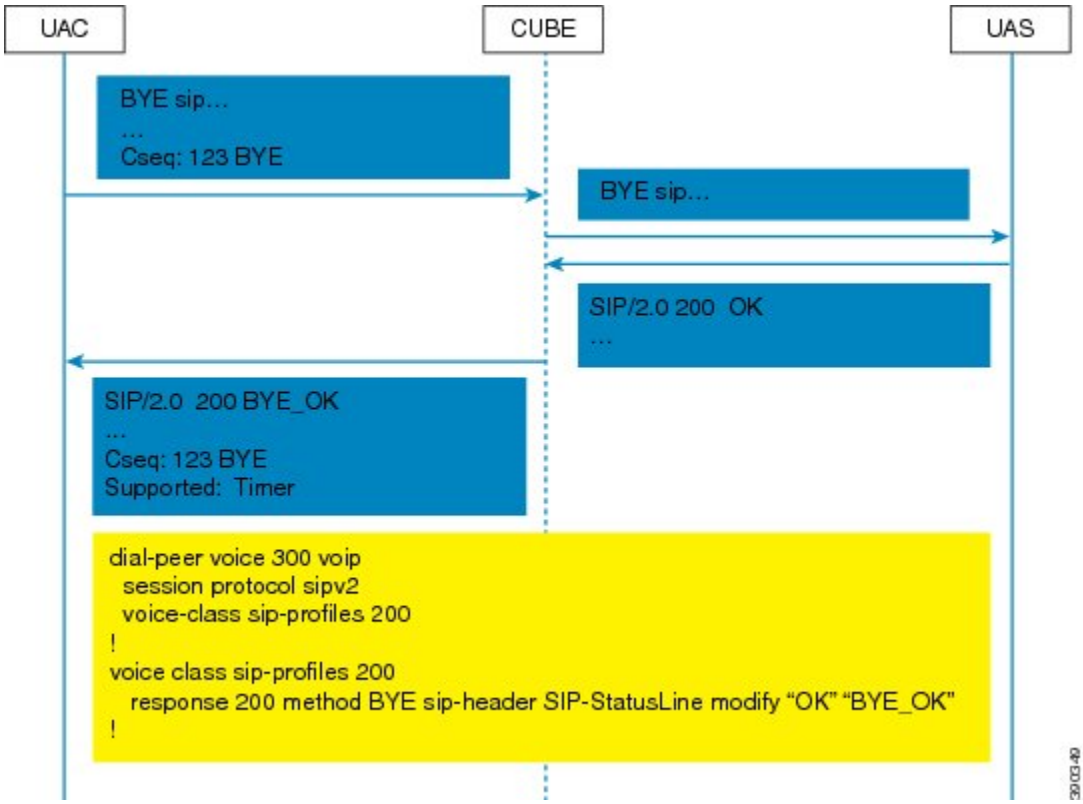
	Command or Action	Purpose
Step 10	**response** *response-code* **peer-header sip SIP-StatusLine copy** *match-pattern copy-variable* **Example:** `Device(config-class)# response ANY peer-header sip SIP-StatusLine copy "(.*)" u01`	Copies responses from the corresponding incoming call leg into a copy variable.
Step 11	**response** *response-code* **sip-header SIP-StatusLine modify** *match-pattern copy-variable* **Example:** `Device(config-class)# response ANY sip-header SIP-StatusLine modify ".*" "\u01"`	Modifies an outgoing response using the copy variable defined in the previous step.
Step 12	**exit** **Example:** `Device(config-class)# exit`	Exits voice class configuration mode and returns to global configuration mode.

What to Do Next

Apply the SIP profile to the outbound dial peer to copy the SIP response to the outbound leg.

Modifying Status-Line Header of Outgoing SIP Response with User Defined Values

Figure 15: Call Flow Configuring a New Status Line for an Outgoing SIP Response Based on an Incoming SIP Request



SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice class sip-profiles** *tag*
4. **response** *response-code* [**method** *method-type*] **sip-header SIP-StatusLine modify** *match-pattern replacement-pattern*
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**voice class sip-profiles** *tag* **Example:** `Device(config)# voice class sip-profiles 10`	Enables dial peer-based VoIP SIP profile configurations and enters voice class configuration mode.
Step 4	**response** *response-code* [**method** *method-type*] **sip-header SIP-StatusLine modify** *match-pattern replacement-pattern* **Example:** Modifying status line of a SIP header to a user-defined response type: `Device(config-class)# response 404 sip-header SIP-StatusLine modify "404 Not Found" "404 MyError"`	Modifies SIP status line of a SIP response with user-defined values.
Step 5	**exit** **Example:** `Device(config-class)# exit`	Exits voice class configuration mode.

What to Do Next

Associate the SIP profile with an outbound dial peer.

Configuration Examples for SIP Profiles

Example: Adding a SIP, SDP, or Peer Header

Example: Adding "b=AS:4000" SDP header to the video-media Header of the INVITE SDP Request Messages

```
Device(config)# voice class sip-profiles 10
```

```
Device(config-class)# request INVITE sdp-header Video-Bandwidth-Info add "b=AS:4000"
Device(config-class)# end
```

Example: Adding the Retry-After Header to the SIP 480 Response Messages

```
Device(config)# voice class sip-profiles 20
Device(config-class)# response 480 sip-header Retry-After add “Retry-After: 60”
Device(config-class)# end
```

Example: Adding "User-Agent: SIP-GW-UA" to the User-Agent Field of the 200 Response SIP Messages

```
Device(config)# voice class sip-profiles 40
Device(config-class)# response 200 sip-header User-Agent add "User-Agent: SIP-GW-UA"
Device(config-class)# end
```

Applying the SIP Profiles

```
! Applying SIP profiles globally
Device(config)# voice service voip
Device (config-voi-serv) sip-profiles 20
Device (config-voi-serv) end

! Applying SIP profiles to one dial peer only
Device (config) dial-peer voice 10 voip
Device (config-dial-peer) voice-class sip profiles 30
Device (config-dial-peer) voice-class sip profiles 40
Device (config-dial-peer) voice-class sip profiles 10
Device (config-dial-peer) end
```

Example: Modifying a SIP, SDP, or Peer Header

Example: Modifying SIP-Req-URI of the Header of the INVITE and RE-INVITE SIP Request Messages to include "user=phone"

```
Device(config)# voice class sip-profiles 30
Device(config-class)# request INVITE sip-header SIP-Req-URI modify "; SIP/2.0" ";user=phone
 SIP/2.0"
Device(config-class)# request RE-INVITE sip-header SIP-Req-URI modify "; SIP/2.0" ";user=phone
 SIP/2.0"
Device(config-class)# end
```

Modify the From Field of a SIP INVITE Request Messages to “gateway@gw-ip-address” Format

For example, modify 2222000020@10.13.24.7 to gateway@10.13.24.7

```
Device(config)# voice class sip-profiles 20
Device(config-class)# request INVITE sip-header From modify "(<.*:)(.*@)" "\1gateway@"
```

Replace "CiscoSystems-SIP-GW-UserAgent" with "-" in the Originator Header of the SDP in INVITE Request Messages

```
Device(config)# voice class sip-profiles 10
Device(config-class)# request INVITE sdp-header Session-Owner modify
"CiscoSystems-SIP-GW-UserAgent“ "-"
```

Convert "sip uri" to "tel uri" in Req-URI, From and To Headers of SIP INVITE Request Messages

For example, modify sip:2222000020@9.13.24.6:5060" to "tel:2222000020

```
Device(config)# voice class sip-profiles 40
Device(config-class)# request INVITE sip-header SIP-Req-URI modify "sip:(.*)@[^ ]+" "tel:\1"
Device(config-class)# request INVITE sip-header From modify "<sip:(.*)@.*>" "<tel:\1>"
Device(config-class)# request INVITE sip-header To modify "<sip:(.*)@.*>" "<tel:\1>"
```

Example: Change the Audio Attribute Ptime:20 to Ptime:30

Inbound ptime:

```
a=ptime:20
```

Outbound ptime:

```
a=ptime:30
Device(config)# voice class sip-profiles 103
Device(config-class)# request ANY sdp-header Audio-Attribute modify "a=ptime:20" "a=ptime:30"
```

Example: Modify Audio direction "Audio-Attribute"

Some service providers or customer equipment reply to delay offer invites and or re-invites that contain a=inactive with a=inactive, a=recvonly, or a=sendonly. This can create an issue when trying to transfer or retrieve a call from hold. The result is normally one-way audio after hold or resume or transfer or moh is not heard. To resolve this issue changing the audio attribute to Sendrecv prevents the provider from replaying back with a=inactive, a=recvonly, or a=sendonly.

Case 1:

```
Inbound Audio-Attribute

a=inactive

Outbound Audio-Attribute

a=sendrecv
```

Case 2:

```
Inbound Audio-Attribute

a=recvonly

Outbound Audio-Attribute

a=sendrecv
```

Case 3

```
Inbound Audio-Attribute

a=sendonly

Outbound Audio-Attribute

a=sendrecv
Device(config)# voice class sip-profiles 104
Device(config-class)# request any sdp-header Audio-Attribute modify "a=inactive" "a=sendrecv"
Device(config-class)# request any sdp-header Audio-Attribute modify "a=recvonly" "a=sendrecv"
Device(config-class)# request any sdp-header Audio-Attribute modify "a=sendonly" "a=sendrecv"

Device(config-class)# response any sdp-header Audio-Attribute modify "a=inactive" "a=sendrecv"
Device(config-class)# response any sdp-header Audio-Attribute modify "a=recvonly" "a=sendrecv"
Device(config-class)# response any sdp-header Audio-Attribute modify "a=sendonly" "a=sendrecv"
```

Applying the SIP Profiles to Dial Peers

```
! Applying SIP Profiles globally
Device(config)# voice service voip
Device (config-voi-serv) sip-profiles 20
Device (config-voi-serv) sip-profiles 10
Device (config-voi-serv) sip-profiles 40
Device (config-voi-serv) sip-profiles 103
Device (config-voi-serv) sip-profiles 104
Device (config-voi-serv) exit

! Applying SIP Profiles to one dial peer only
Device (config) dial-peer voice 90 voip
Device (config-dial-peer) voice-class sip profiles 30
```

Example: Remove a SIP, SDP, or Peer Header

Remove Cisco-Guid SIP header from all Requests and Responses

```
Device(config)# voice class sip-profiles 20
Device(config-class)# request ANY sip-header Cisco-Guid remove
Device(config-class)# response ANY sip-header Cisco-Guid remove
Device(config-class)# end
```

Remove Server Header from 100 and 180 SIP Response Messages

```
Device(config)# voice class sip-profiles 20
Device(config-class)# response 100 sip-header Server remove
Device(config-class)# response 180 sip-header Server remove
Device(config-class)# end
```

Example: Modifying Diversion Headers

Example: Modify Diversion Headers from Three-Digit Extensions to Ten Digits.

Most service providers require a ten digit diversion header. Prior to Call manager 8.6, Call manager would only send the extension in the diversion header. A SIP profile can be used to make the diversion header ten digits.

Call manager version 8.6 and above has the field “Redirecting Party Transformation CSS” which lets you expand the diversion header on the call manager.

The SIP profile will look for a diversion header containing "<sip:5..." , where ... stands for the three-digit extension and then concatenates 9789365 with these three digits.

Original Diversion Header:

```
Diversion:<sip:5100@161.44.77.193>;privacy=off;reason=unconditional;counter=1;screen=no
```

Modified Diversion Header:

```
Diversion: <sip:9789365100@10.86.176.19>;privacy=off;reason=unconditional;counter=1;screen=no
```

```
Device(config)# voice class sip-profiles 101
Device(config-class)# request Invite sip-header Diversion modify "<sip:5(...)@"
"<sip:9789365\1@"
Device(config-class)# end
```

Example: Create a Diversion header depending on the area code in the From field

Most service providers require a redirected call to have a diversion header that contains a full 10 digit number that is associated with a SIP trunk group. Sometimes, a SIP trunk may cover several different area codes, states, and geographic locations. In this scenario, the service provider may require a specific number to be placed in the diversion header depending on the calling party number.

In the below example, if the From field has an area code of 978 "<sip:978", the SIP profile leaves the From field as is and adds a diversion header.

```
Device(config)# voice class sip-profiles 102
Device(config-class)# request INVITE sip-header From modify "From:(.*)<sip:978(.*)@(.*)"
"From:\1<sip:978\2@\3\x0ADiversion:
<sip:9789365000@10.86.176.19:5060;privacy=off;reason=unconditional;counter=1;screen=no"
```

The below diversion header is added. There was no diversion header before this was added:

```
Diversion: <sip:9789365000@10.86.176.19:5060;transport=udp>"
```

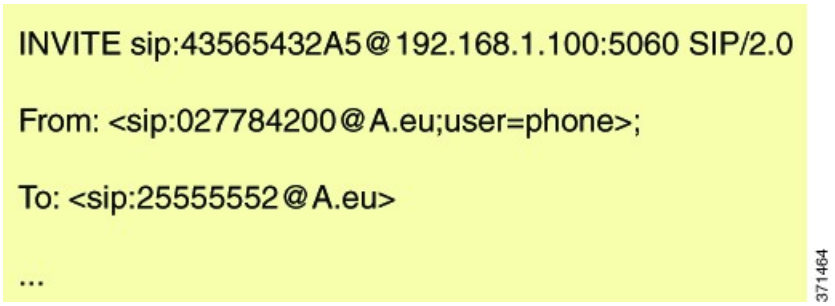
Example: Copying the To Header into the SIP-Req-URI

Copying Contents from One Header to Another

Given below is a scenario in an organization, where the provider has sent only a global reference number in the SIP-Req-URI header of the INVITE message, and has placed the actual phone destination number only in the To: SIP header. The CUCM typically routes on the SIP-Req-URI.

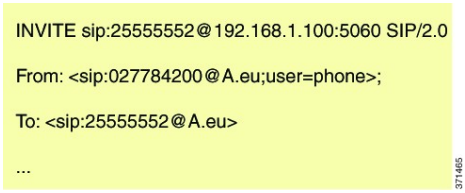
Given below is the original SIP message, where the INVITE has a non-routable value of 43565432A5. The actual phone destination number is 25555552 and is present in the To: SIP header.

Figure 16: Incoming SIP Message



Given below is the SIP message that is required. Note that 43565432A5 has changed to 25555552 in the SIP INVITE.

Figure 17: Modified SIP Message



Because CUBE is a back-to-back user agent, the incoming dial peer is matched to the outgoing dial peer. The SIP Profile configured below copies the value from the incoming dial peer

```
Device# voice class sip-profiles 1

!Copy the To header from the incoming dial peer into variable u01
Device(config-class)# request INVITE peer-header sip TO copy "sip:(.*)@" u01

!Modify the outgoing SIP Invite with this variable.
Device(config-class)# request INVITE sip-header SIP-Req-URI modify ".*@(.*)" "INVITE
sip:\u01@\1"
```

Apply the SIP profile to the incoming dial peer.

```
Device(config)# dial-peer voice 99 voip
Device(config-dial-peer)# outgoing to CUCM
Device(config-dial-peer)# destination-pattern 02555555.
Device(config-dial-peer)# session protocol sipv2
Device(config-dial-peer)# session target ipv4:10.1.2.3

!Applying SIP profile to the dial peer
Device(config-dial-peer)# voice-class sip profiles 1
Device(config-dial-peer)# voice-class code 1
Device(config-dial-peer)# dtmf-relay rtp-nte
Device(config-dial-peer)# no vad
```

Additionally, if you would like to copy the To: Header from the inbound dial peer to the outbound dial peer, use a copy list.

```
!Create a copy List
Device(config)# voice class sip-copylist 1
Device(config-class)# sip-header TO
Device(config-class)# exit

!Apply the copy list to incoming dial peer.
Device(config)# dial-peer voice 1 voip
Device(config-dial-peer)# description incoming SIP Trunk
Device(config-dial-peer)# session protocol sipv2
Device(config-dial-peer)# session target sip-server
Device(config-dial-peer)# incoming uri to TRUNK
Device(config-dial-peer)# voice-class code 1
Device(config-dial-peer)# voice-class sip copy-list 1

Device(config)# voice class uri TRUNK sip
Device(config-class)# user-id 2555555.
Device(config-class)# end
```

Example: Passing a Header Not Supported by CUBE

CUBE does not pass “x-cisco-tip”. However, certain TelePresence equipments require “TIP”.

The SIP profile below will look for "x-cisco-tip" in the inbound contact header then pass it in the outbound contact header.

Inbound Contact Header

```
Contact: <sip:89016442998@161.44.77.193;transport=udp>;x-cisco-tip
```

Outbound Contact Header

```
Contact: <sip:89016442998@10.86.176.19:5060>;x-cisco-tip
```

Create a copylist to pass the Contact Header from the incoming message to the outgoing message. The “x-cisco-tip” is not copied in this step as it is unsupported by CUBE.

```
!Create a copyList
Device(config)# voice class sip-copylist 1
Device(config-class)# sip-header Contact
Device(config-class)# exit

!Apply the copylist to incoming dial peer.
Device(config)# dial-peer voice 1 voip
Device(config-dial-peer)# description incoming SIP Trunk
Device(config-dial-peer)# incoming called-number
Device(config-dial-peer)# voice-class sip copy-list 1
```

Create a SIP profile that copies “x-cisco-tip” into a variable, and use that variable to modify the outgoing Contact header. Apply the SIP profile to an outbound dial peer.

```
Device# voice class sip-profiles 3001

!Copy the Contact header from the incoming dial peer into variable u01
Device(config-class)# request INVITE peer-header sip Contact copy "(;x-cisco-tip)" u01

!Modify the outgoing SIP Invite with this variable.
Device(config-class)#  request INVITE sip-header Contact modify "$" "\u01"″


!Apply the SIP Profile to the outgoing dial peer.
Device(config)# dial-peer voice 5000 voip
Device(config-dial-peer)# description outbound SIP
Device(config-dial-peer)# destination-pattern 5...$
Device(config-dial-peer)# voice-class sip profiles 3001
```

Example: Sample SIP Profile Application on SIP Invite Message

The SIP profile configured is below:

```
voice class sip-profiles 1
  request INVITE sdp-header Audio-Bandwidth-Info add "b=AS:1600“
  request ANY sip-header Cisco-Guid remove
  request INVITE sdp-header Session-Owner modify "CiscoSystems-SIP-GW-UserAgent" "-“
```

The SIP INVITE message before the SIP profile has been applied is show below:

```
INVITE sip:2222000020@9.13.40.250:5060 SIP/2.0
Via: SIP/2.0/UDP 9.13.40.249:5060;branch=z9hG4bK1A203F
From: "sipp " <sip:1111000010@9.13.40.249>;tag=F11AE0-1D8D
To: <sip:2222000020@9.13.40.250>
Date: Mon, 29 Oct 2007 19:02:04 GMT
Call-ID: 4561B116-858811DC-804DEF2E-4CF2D71B@9.13.40.249
Cisco-Guid: 1163870326-2240287196-2152197934-1290983195
Content-Length: 290
```

```
v=0
o=CiscoSystemsSIP-GW-UserAgent 6906 8069 IN IP4 9.13.40.249
s=SIP Call
c=IN IP4 9.13.40.249
t=0 0
m=audio 17070 RTP/AVP 0
c=IN IP4 9.13.40.249
a=rtpmap:0 PCMU/8000
a=ptime:20
```

The SIP INVITE message after the SIP profile has been applied is shown below:

- The Cisco-Guid has been removed.
- CiscoSystemsSIP-GW-UserAgent has been replaced with -.
- The Audio-Bandwidth SDP header has been added with the value b=AS:1600.

```
INVITE sip:2222000020@9.13.40.250:5060 SIP/2.0
Via: SIP/2.0/UDP 9.13.40.249:5060;branch=z9hG4bK1A203F
From: "sipp " <sip:1111000010@9.13.40.249>;tag=F11AE0-1D8D
To: <sip:2222000020@9.13.40.250>
Date: Mon, 29 Oct 2007 19:02:04 GMT
Call-ID: 4561B116-858811DC-804DEF2E-4CF2D71B@9.13.40.249
Content-Length: 279

v=0
o=- 6906 8069 IN IP4 9.13.40.249
s=SIP Call
c=IN IP4 9.13.40.249
t=0 0
m=audio 17070 RTP/AVP 0
c=IN IP4 9.13.40.249
a=rtpmap:0 PCMU/8000
a=ptime:20
b=AS:1600
```

Feature Information for Configuring SIP Profiles

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

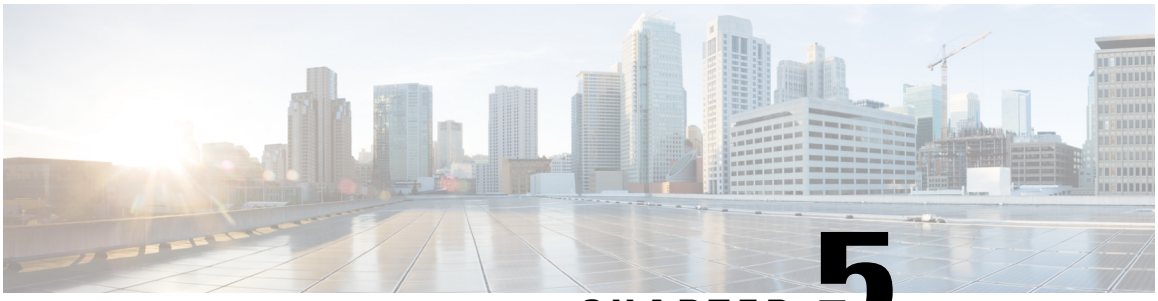
Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 7: Feature Information for Configuring SIP Profiles

Feature Name	Releases	Feature Information
SIP Profiles (for inbound messages)	15.4(2)T Cisco IOS XE Release 3.12S	This feature extends support to inbound messages. This feature modifies the following commands: The **inbound** keyword was added to the **sip-profiles** and **voice-class sip profiles** commands.

Feature Name	Releases	Feature Information
SIP Profile Enhancements for SIP responses and error codes	15.4(1)T Cisco IOS XE Release 3.11S	This feature extends SIP profiles to allow the folloing: • Modification of the outgoing SIP response status line. Previously, only modification of outgoing SIP requests and responses was possible. • Copying of the incoming SIP response status-line. The information from the peer-leg status-line can then be copied to user-variables and applied to the outbound response status-line. This option can be used to pass-thru the error-code and error phrase from peer-leg. Previously, only copying of SIP headers were possible. • Before applying a SIP profile to a response from CUBE, the response can be mapped to its corresponding request.
Support for Rotary calls and Media Forking	15.3(1)T	With CSCty41575, this feature was enhanced to support forked and rotary calls.
Configuring SIP Profiles (Copy)	15.1(3)T Cisco IOS XE Release 3.6S	This feature allows users to copy content from one header to the another. This is done by copying the content of messages into variables which can then be used to modify other SIP headers. This feature modifies the following commands: **voice class sip-profiles**, **response**, **request**, **voice-class sip copy-list**, **sip-header**

Feature Name	Releases	Feature Information
Configuring SIP Profile (Add, Delete or Modify)	12.4(15)XZ 12.4(20)T Cisco IOS XE Release 2.5	This feature allows users to change (add, delete, or modify) the standard SIP messages that are sent or received for better interworking with different SIP entities. This feature introduces the following commands: **voice class sip-profiles**, **response**, **request**.



CHAPTER 5

Dial Peer Matching

CUBE allows VoIP-to-VoIP connection by routing calls from one VoIP dial peer to another. As VoIP dial peers can be handled by either SIP or H.323, CUBE can be used to interconnect VoIP networks of different signaling protocols. VoIP internetworking is achieved by connecting an inbound dial peer with an outbound dial peer.

Dial Peers in CUBE

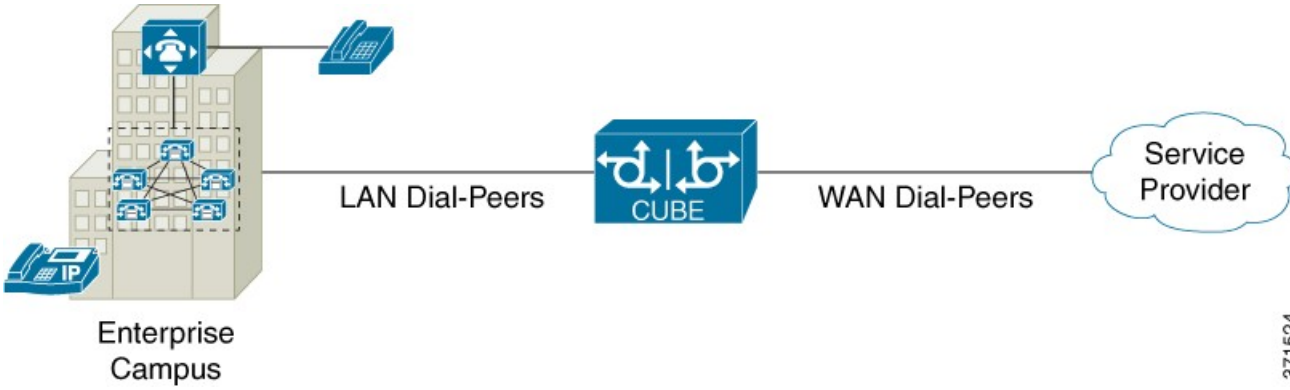
A dial peer is a static routing table mapping phone numbers to interfaces or IP addresses

A call leg is a logical connection between two routers or between a router and a VoIP endpoint. A dial peer is associated or matched to each call leg according to attributes that define a packet switched network, like destination address.

Voice-network dial peers are matched to call legs based on configured parameters, after which an outbound dial peer is provisioned to an external component using the component's IP address. For more information, refer to Dial Peer Configuration Guide.

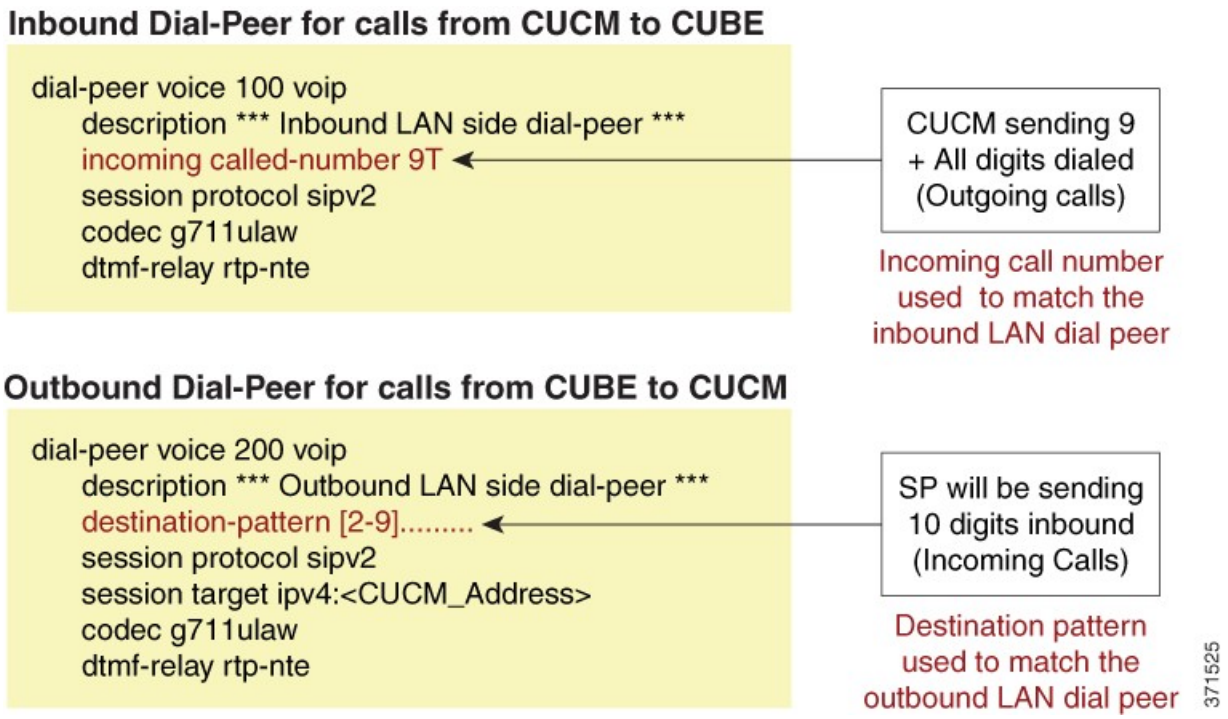
In CUBE, dial peers can also be classified as LAN dial peers and WAN dial peers based on the connecting entity that CUBE sends or receives calls from.

Figure 18: LAN and WAN Dial Peers



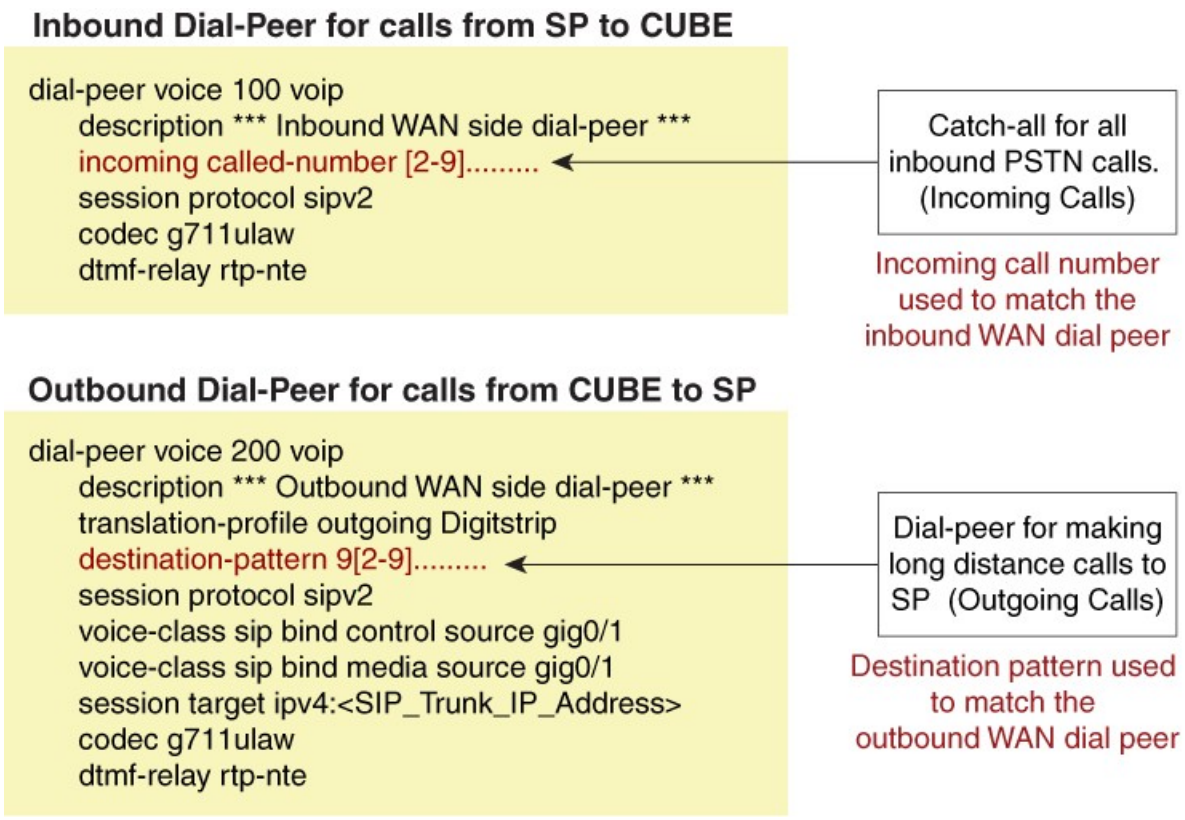
A LAN dial peer is used to send or receive calls between CUBE and the Private Branch Exchange (PBX), a system of telephone extensions within an enterprise. Given below are examples of inbound and outbound LAN dial peers.

Figure 19: LAN Dial Peers



A WAN dial peer is used to send or receive calls between CUBE and the SIP trunk provider. Given below are examples of inbound and outbound WAN dial peers.

Figure 20: WAN Dial Peers



Configuring Inbound and Outbound Dial Peers Matching for CUBE

The following commands can be used for inbound and outbound dial peer matching in CUBE

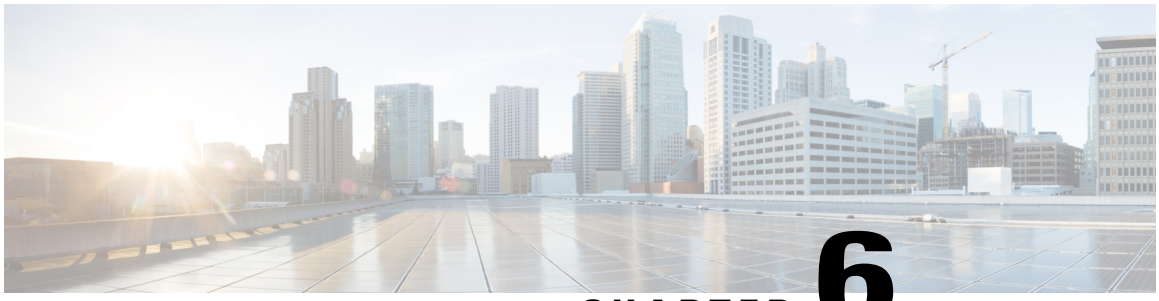
Table 8: Incoming Dial Peer Matching

Command in Dial-Peer Configuration	Description	Call Setup Element
incoming called-number *DNIS_string*	This command uses the destination number that was called to match the incoming call leg to an inbound dial peer. This number is called the dialed number identification service (DNIS) number.	DNIS number

Command in Dial-Peer Configuration	Description	Call Setup Element
answer-address *ANI_string*	This command uses the calling number to match the incoming call leg to an inbound dial peer. This number is called the originating calling number or automatic number identification (ANI) string.	ANI string
destination-pattern *ANI_string*	This command uses the inbound call leg to the inbound dial peer.	ANI string for inbound
{**incoming called | incoming calling**} **e164-pattern-map** *pattern-map-group-id*	This command uses a group of incoming called (DNIS) or incoming calling (ANI) number patterns to match the inbound call leg to an inbound dial peer. The command calls a globally defined voice class identifier where the E.164 patter groups are configured.	E.164 Patterns
incoming uri {**from | request | to | via**} *URI_class_identifier*	This command uses the directory URI (Uniform Resource Identifier) number to match the outgoing SIP call leg to an outgoing dial peer. This directory URI is part of the SIP address of a device. The command calls a globally defined voice class identifier where the directory URI is configured. It requires the configuration of **session protocol sipv2**	Directory URI
incoming uri {**called | callling**} *URI_class_identifier*	This command uses the directory URI (Uniform Resource Identifier) number to match the outgoing H.323 call leg to an outgoing dial peer. The command calls a globally defined voice class identifier where the directory URI is configured.	Directory URI

Table 9: Outgoing Dial Peer Matching

Dial Peer Command	Description	Call Setup Element
destination-pattern *DNIS_string*	This command uses DNIS string to match the outbound call leg to the outbound dial peer.	DNIS string for outbound ANI string for inbound
destination *URI_class_identifier*	This command uses the directory URI (Uniform Resource Identifier) number to match the outgoing call leg to an outgoing dial peer. This directory URI is part of the SIP address of a device. The command actually refers to a globally defined voice class identifier where the directory URI is configured.	Directory URI
destination e164-pattern-map *pattern-map-group-id*	This command uses a group of destination number patterns to match the outbound call leg to an outbound dial peer. The command calls a globally defined voice class identifier where the E.164 pattern groups are configured.	E.164 patterns



CHAPTER 6

Additional References

The following sections provide references related to the Cisco Unified Border Element (Enterprise) Configuration Guide.

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Cisco IOS Voice commands	*Cisco IOS Voice Command Reference*
Cisco IOS Voice Configuration Library	For more information about Cisco IOS voice features, including feature documents, and troubleshooting information--at http://www.cisco.com/en/US/docs/ios/12_3/vvf_c/cisco_ios_voice_configuration_library_glossary/vcl.htm
Cisco IOS Release 15.0	Cisco IOS Release 15.0 Configuration Guides
Cisco IOS Release 12.2	Cisco IOS Voice, Video, and Fax Configuration Guide, Release 12.2

Related Topic	Document Title
internet Low Bitrate Codec (iLBC) Documents	• Codecs section of the Dial Peer Configuration on Voice Gateway Routers Guide http://www.cisco.com/en/US/docs/ios-xml/ios/voice/dialpeer/configuration/15-mt/vd-dp-overview.html • Dial Peer Features and Configuration section of the Dial Peer Configuration on Voice Gateway Routers Guide http://www.cisco.com/en/US/docs/ios-xml/ios/voice/dialpeer/configuration/15-mt/vd-dp-feat-cfg.html
Related Application Guides	• *Cisco Unified Communications Manager and Cisco IOS Interoperability Guide* • *Cisco IOS SIP Configuration Guide* • Cisco Unified Communications Manager (CallManager) Programming Guides
Troubleshooting and Debugging guides	• Cisco IOS Debug Command Reference, Release 12.4. • *Troubleshooting and Debugging VoIP Call Basics* at http://www.cisco.com/en/US/tech/tk1077/technologies_tech_note09186a0080094045.shtml • *VoIP Debug Commands* at http://www.cisco.com/en/US/docs/routers/access/1700/1750/software/configuration/guide/debug.html

Standards

Standard	Title
ITU-T G.711	--

MIBs

MIB	MIBs Link
• CISCO-PROCESS MIB • CISCO-MEMORY-POOL-MIB • CISCO-SIP-UA-MIB • DIAL-CONTROL-MIB • CISCO-VOICE-DIAL-CONTROL-MIB • CISCO-DSP-MGMT-MIB • IF-MIB • IP-TAP-MIB • TAP2-MIB • USER-CONNECTION-TAP-MIB	To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

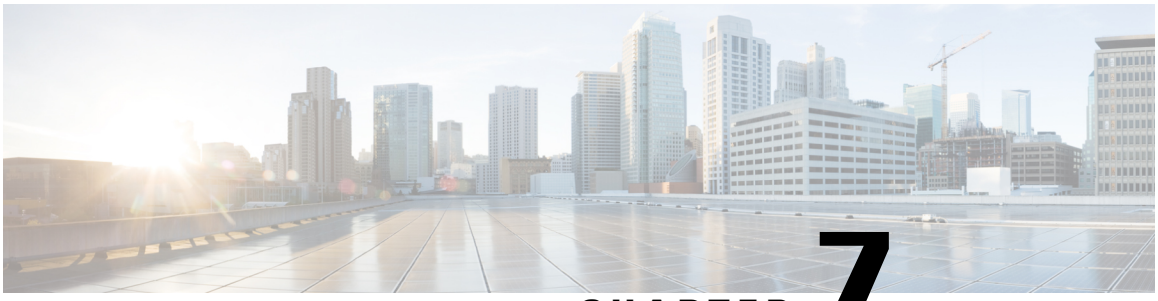
RFCs

RFC	Title
RFC 1889	*RTP: A Transport Protocol for Real-Time Applications*
RFC 2131	*Dynamic Host Configuration Protocol*
RFC 2132	*DHCP Options and BOOTP Vendor Extensions*
RFC 2198	*RTP Payload for Redundant Audio Data*
RFC 2327	*SDP: Session Description Protocol*
RFC 2543	*SIP: Session Initiation Protocol*
RFC 2543-bis-04	*SIP: Session Initiation Protocol, draft-ietf-sip-rfc2543bis-04.txt*
RFC 2782	*A DNS RR for Specifying the Location of Services (DNS SRV)*
RFC 2833	*RTP Payload for DTMF Digits, Telephony Tones and Telephony Signals*

RFC	Title
RFC 3203	*DHCP reconfigure extension*
RFC 3261	*SIP: Session Initiation Protocol*
RFC 3262	*Reliability of Provisional Responses in Session Initiation Protocol (SIP)*
RFC 3323	*A Privacy Mechanism for the Session Initiation Protocol (SIP)*
RFC 3325	*Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks*
RFC 3515	*The Session Initiation Protocol (SIP) Refer Method*
RFC 3361	*Dynamic Host Configuration Protocol (DHCP-for-IPv4) Option for Session Initiation Protocol (SIP) Servers*
RFC 3455	*Private Header (P-Header) Extensions to the Session Initiation Protocol (SIP) for the 3rd-Generation Partnership Project (3GPP)*
RFC 3608	*Session Initiation Protocol (SIP) Extension Header Field for Service Route Discovery During Registration*
RFC 3711	*The Secure Real-time Transport Protocol (SRTP)*
RFC 3925	Vendor-Identifying Vendor Options for Dynamic Host Configuration Protocol version 4 (DHCPv4)

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html



CHAPTER 7

Glossary

Glossary

AMR-NB —Adaptive Multi Rate codec - Narrow Band.

Allow header —Lists the set of methods supported by the UA generating the message.

bind — In SIP, configuring the source address for signaling and media packets to the IP address of a specific interface.

call —In SIP, a call consists of all participants in a conference invited by a common source. A SIP call is identified by a globally unique call identifier. A point-to-point IP telephony conversation maps into a single SIP call.

call leg —A logical connection between the router and another endpoint.

CLI —command-line interface.

Content-Type header —Specifies the media type of the message body.

CSeq header —Serves as a way to identify and order transactions. It consists of a sequence number and a method. It uniquely identifies transactions and differentiates between new requests and request retransmissions.

delta —An incremental value. In this case, the delta is the difference between the current time and the time when the response occurred.

dial peer —An addressable call endpoint.

DNS -—Domain Name System. Used to translate H.323 IDs, URLs, or e-mail IDs to IP addresses. DNS is also used to assist in locating remote gatekeepers and to reverse-map raw IP addresses to host names of administrative domains.

DNS SRV —Domain Name System Server. Used to locate servers for a given service.

DSP —Digital Signal Processor.

DTMF —dual-tone multifrequency. Use of two simultaneous voice-band tones for dialing (such as touch-tone).

EFXS —IP phone virtual voice ports.

FQDN —fully qualified domain name. Complete domain name including the host portion; for example, *serverA.companyA.com* .

FXS —analog telephone voice ports.

gateway —A gateway allows SIP or H.323 terminals to communicate with terminals configured to other protocols by converting protocols. A gateway is the point where a circuit-switched call is encoded and repackaged into IP packets.

H.323 —An International Telecommunication Union (ITU-T) standard that describes packet-based video, audio, and data conferencing. H.323 is an umbrella standard that describes the architecture of the conferencing system and refers to a set of other standards (H.245, H.225.0, and Q.931) to describe its actual protocol.

iLBC —internet Low Bitrate Codec.

INVITE**—**A SIP message that initiates a SIP session. It indicates that a user is invited to participate, provides a session description, indicates the type of media, and provides insight regarding the capabilities of the called and calling parties.

IP**—**Internet Protocol. A connectionless protocol that operates at the network layer (Layer 3) of the OSI model. IP provides features for addressing, type-of-service specification, fragmentation and reassemble, and security. Defined in RFC 791. This protocol works with TCP and is usually identified as TCP/IP. See TCP/IP.

ISDN —Integrated Services Digital Network.

Minimum Timer —Configured minimum value for session interval accepted by SIP elements (proxy, UAC, UAS). This value helps minimize the processing load from numerous INVITE requests.

Min-SE —Minimum Session Expiration. The minimum value for session expiration.

multicast —A process of transmitting PDUs from one source to many destinations. The actual mechanism (that is, IP multicast, multi-unicast, and so forth) for this process might be different for LAN technologies.

originator —User agent that initiates the transfer or Refer request with the recipient.

PDU —protocol data units. Used by bridges to transfer connectivity information.

PER —Packed Encoding Rule.

proxy —A SIP UAC or UAS that forwards requests and responses on behalf of another SIP UAC or UAS.

proxy server —An intermediary program that acts as both a server and a client for the purpose of making requests on behalf of other clients. Requests are serviced internally or by passing them on, possibly after translation, to other servers. A proxy interprets and, if necessary, rewrites a request message before forwarding it.

recipient —User agent that receives the Refer request from the originator and is transferred to the final recipient.

redirect server —A server that accepts a SIP request, maps the address into zero or more new addresses, and returns these addresses to the client. It does not initiate its own SIP request or accept calls.

re-INVITE —An INVITE request sent during an active call leg.

Request URI —Request Uniform Resource Identifier. It can be a SIP or general URL and indicates the user or service to which the request is being addressed.

RFC —Request For Comments.

RTP —Real-Time Transport Protocol (RFC 1889)

SCCP —Skinny Client Control Protocol.

SDP**—**Session Description Protocol. Messages containing capabilities information that are exchanged between gateways.

session —A SIP session is a set of multimedia senders and receivers and the data streams flowing between the senders and receivers. A SIP multimedia conference is an example of a session. The called party can be invited several times by different calls to the same session.

session expiration —The time at which an element considers the call timed out if no successful INVITE transaction occurs first.

session interval —The largest amount of time that can occur between INVITE requests in a call before a call is timed out. The session interval is conveyed in the Session-Expires header. The UAS obtains this value from the Session-Expires header of a 2*xx* INVITE response that it sends. Proxies and UACs determine this value from the Session-Expires header in a 2*xx* INVITE response they receive.

SIP —Session Initiation Protocol. An application-layer protocol originally developed by the Multiparty Multimedia Session Control (MMUSIC) working group of the Internet Engineering Task Force (IETF). Their goal was to equip platforms to signal the setup of voice and multimedia calls over IP networks. SIP features are compliant with IETF RFC 2543, published in March 1999.

SIP URL —Session Initiation Protocol Uniform Resource Locator. Used in SIP messages to indicate the originator, recipient, and destination of the SIP request. Takes the basic form of *user@host* , where *user* is a name or telephone number, and *host* is a domain name or network address.

SPI —service provider interface.

socket listener —Software provided by a socket client to receives datagrams addressed to the socket.

stateful proxy —A proxy in keepalive mode that remembers incoming and outgoing requests.

TCP —Transmission Control Protocol. Connection-oriented transport layer protocol that provides reliable full-duplex data transmissions. TCP is part of the TCP/IP protocol stack. See also TCP/IP and IP.

TDM —time-division multiplexing.

UA —user agent. A combination of UAS and UAC that initiates and receives calls. See **UAS**and **UAC**.

UAC —user agent client. A client application that initiates a SIP request.

UAS —user agent server. A server application that contacts the user when a SIP request is received and then returns a response on behalf of the user. The response accepts, rejects, or redirects the request.

UDP —User Datagram Protocol. Connectionless transport layer protocol in the TCP/IP protocol stack. UDP is a simple protocol that exchanges datagrams without acknowledgments or guaranteed delivery, requiring that error processing and retransmission be handled by other protocols. UDP is defined in RFC-768.

URI —Uniform Resource Identifier. Takes a form similar to an e-mail address. It indicates the user's SIP identity and is used for redirection of SIP messages.

URL —Universal Resource Locator. Standard address of any resource on the Internet that is part of the World Wide Web (WWW).

User Agent —A combination of UAS and UAC that initiates and receives calls. See **UAS and UAC.**

VFC —Voice Feature Card.

VoIP —Voice over IP. The ability to carry normal telephone-style voice over an IP-based Internet with POTS-like functionality, reliability, and voice quality. VoIP is a blanket term that generally refers to the Cisco standards-based approach (for example, H.323) to IP voice traffic.

Glossary