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 1: 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.

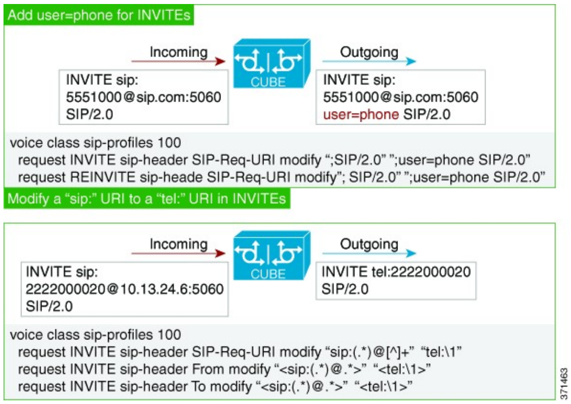
Information About SIP Profile

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 2: 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 22, Example: Modifying a SIP, SDP, or Peer Header, on page 23, and Example: Remove a SIP, SDP, or Peer Header, on page 25 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 22, Example: Modifying a SIP, SDP, or Peer Header, on page 23, and Example: Remove a SIP, SDP, or Peer Header, on page 25 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**.

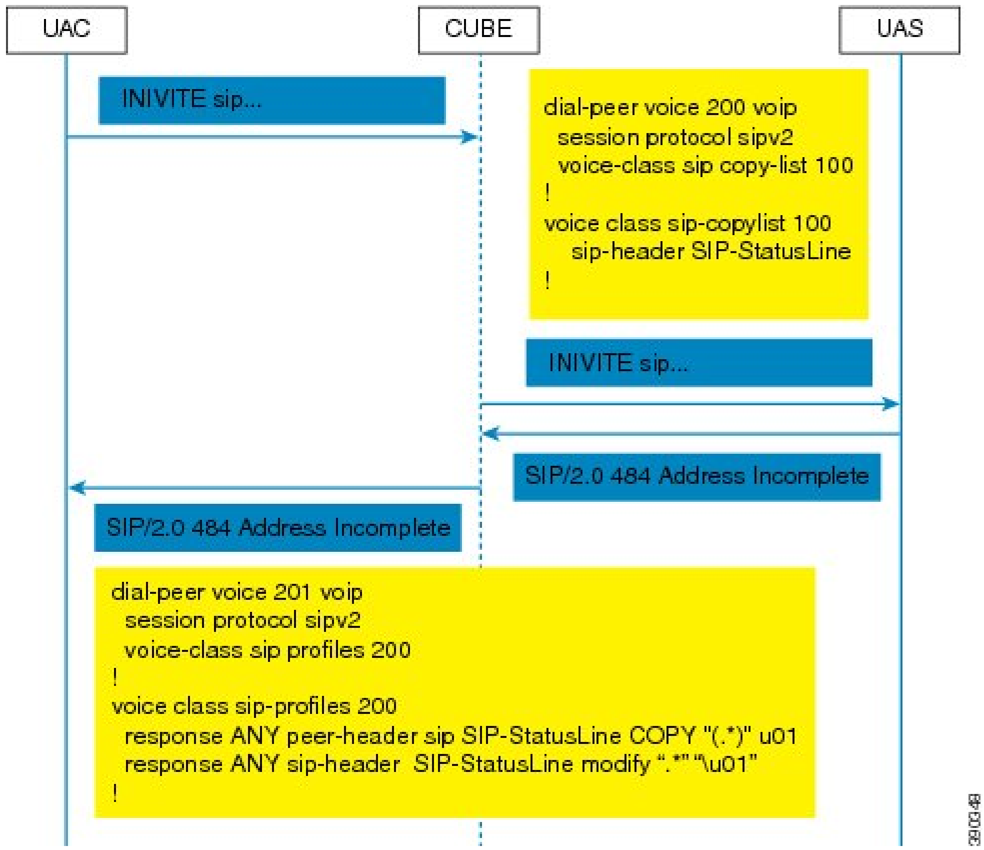
You can copying the SIP response status-lines from one leg to another with the following steps

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 3: 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.

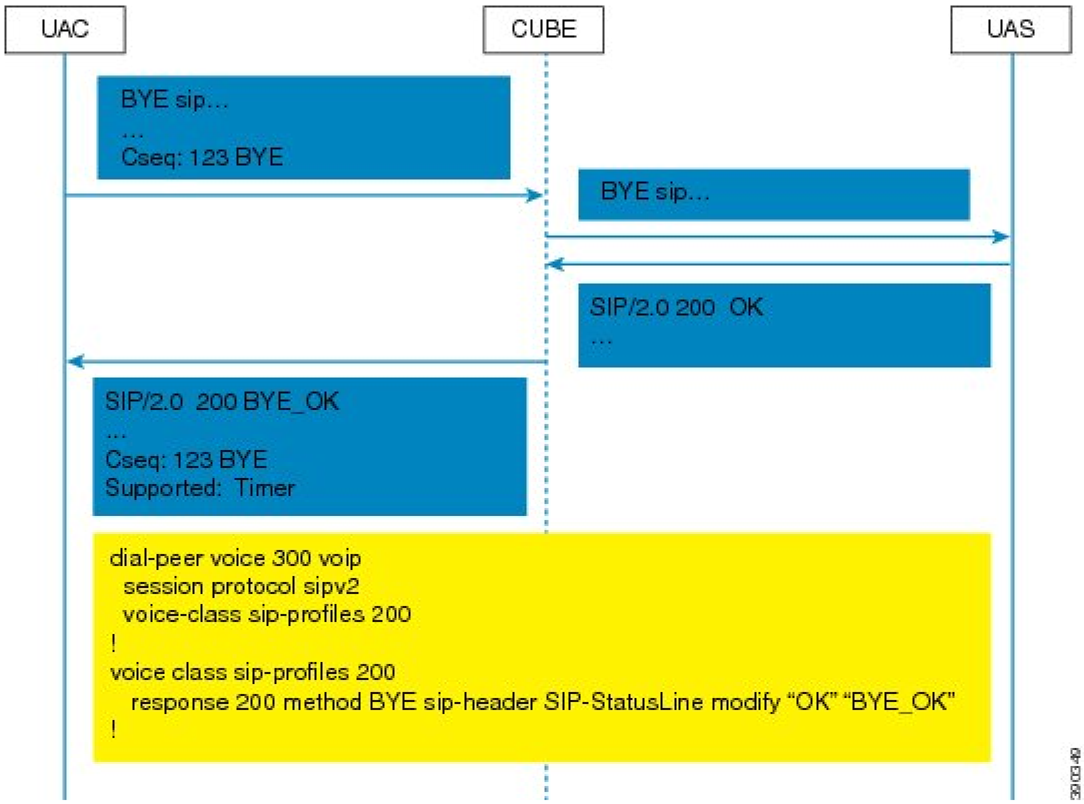
	Command or Action	Purpose
Step 10	**response** *response-code* **peer-header sip SIP-StatusLine copy** *match-pattern copy-variable* **Example:** `Device(config-class)# response ANY peer-header sip SIP-StatusLine copy "(.*)" u01`	Copies responses from the corresponding incoming call leg into a copy variable.
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 4: 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>"
```

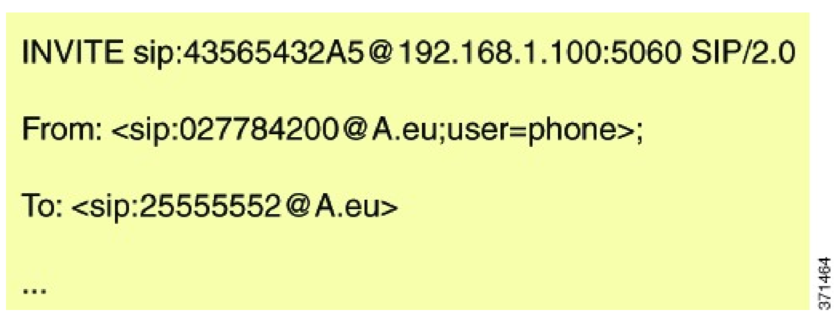
Example: Copying the To Header into the SIP-Req-URI

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 5: Incoming SIP Message



INVITE sip:43565432A5@192.168.1.100:5060 SIP/2.0

From: <sip:027784200@A.eu;user=phone>;

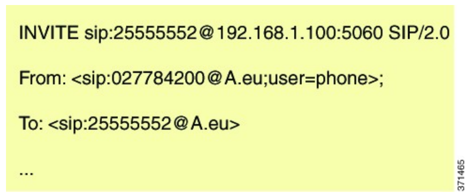
To: <sip:25555552@A.eu>

...

371464

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

Figure 6: 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:222200002O@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:222200002O@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 1: 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**.