Enterprise Application Interoperability for H.323-to-SIP and SIP-to-SIP 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

Enterprise Application Interoperability for H.323-to-SIP and SIP-to-SIP

This Cisco Unified Border Element (Enterprise) is a special Cisco IOS XE software image that runs on Cisco ASR1000. It provides a network-to-network interface point for billing, security, call admission control, quality of service, and signaling interworking. This chapter describes basic gateway functionality, software images, topology, and summarizes supported features.

Note Cisco Product Authorization Key (PAK)--A Product Authorization Key (PAK) is required to configure some of the features described in this guide. Before you start the configuration process, please register your products and activate your PAK at the following URL http://www.cisco.com/go/license .

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.

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

Configuration of Enterprise Application Interoperability for H.323-to-SIP and SIP-to-SIP Features

This chapter contains the following configuration topics:

Cisco UBE (Enterprise) Prerequisites and Restrictions

- Prerequisites for Cisco Unified Border Element (Enterprise)
- Restrictions for Cisco Unified Border Element (Enterprise)

CUCM Interworking

- Cisco Interoperability Portal

www.cisco.com/go/interoperability

Third Party PBX Interworking

- Cisco Interoperability Portal

www.cisco.com/go/interoperability

Application specific interworking notes

- Support for SIP 181 "call is being forwarded" message
- Support for Expires timer reset on receiving or sending SIP 183 message

CHAPTER 2

Configuring SIP 181 Call is Being Forwarded Message

You can configure support for SIP 181 Call is Being Forwarded messages either globally or on a specific dial-peer. Use the **block** command in voice service SIP configuration mode to globally configure Cisco IOS voice gateways and Cisco UBEs to drop specified SIP provisional response messages. To configure settings for an individual dial peer, use the **voice-class sip block** command in dial peer voice configuration mode. Both globally and at the dial peer level, you can also use the **sdp** keyword to further control when the specified SIP message is dropped based on either the absence or presence of SDP information.

Additionally, you can use commands introduced for this feature to configure a Cisco UBE, either globally or at the dial peer level, to map specific received SIP provisional response messages to a different SIP provisional response message on the outgoing SIP dial peer. To do so, use the **map resp-code** command in voice service SIP configuration mode for global configuration or, to configure a specific dial peer, use the **voice-class sip map resp-code** in dial peer voice configuration mode.

This section contains the following tasks:

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.

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

Prerequisites for SIP 181 Call is Being Forwarded Message

Cisco Unified Border Element

Cisco IOS Release 15.0(1)XA or a later release must be installed and running on your Cisco Unified Border Element.

Cisco Unified Border Element (Enterprise)

- Cisco IOS XE Release 3.1S or a later release must be installed and running on your Cisco ASR 1000 Series Router.

Configuring SIP 181 Call is Being Forwarded Message Globally

Perform this task to configure support for SIP 181 messages at a global level in SIP configuration (conf-serv-sip) mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **sip**
5. **block** {**180** | **181** | **183**} [**sdp** {**absent** | **present**}]
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode, or other security level set by a system administrator. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice service voip** **Example:** `Router(config)# voice service voip`	Enters voice service VoIP configuration mode.

	Command or Action	Purpose
Step 4	**sip** **Example:** `Router(conf-voi-serv)# sip`	Enters SIP configuration mode.
Step 5	**block** {**180** | **181** | **183**} [**sdp** {**absent** | **present**}] **Example:** `Router(conf-serv-sip)# block 181 sdp present`	Configures support of SIP 181 messages globally so that messages are passed as is. The sdp keyword is optional and allows for dropping or passing of SIP 181 messages based on the presence or absence of SDP.
Step 6	**exit** **Example:** `Router(conf-serv-sip)# exit`	Exits the current mode.

Configuring SIP 181 Call is Being Forwarded Message at the Dial-Peer Level

Perform this task to configure support for SIP 181 messages at the dial-peer level, in dial peer voice configuration (config-dial-peer) mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag* **voip**
4. **voice-class sip block** {**180** | **181** | **183**} [**sdp** {**absent** | **present**}]
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode, or other security level set by a system administrator. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**dial-peer voice** *tag* **voip** **Example:**	Enters dial peer VoIP configuration mode.

	Command or Action	Purpose
	`Router(config)# dial-peer voice 2 voip`	
Step 4	**voice-class sip block** {**180** | **181** | **183**} [**sdp** {**absent** | **present**}] **Example:** `Router(config-dial-peer)# voice-class sip block 181 sdp present`	Configures support of SIP 181 messages on a specific dial peer so that messages are passed as is. The sdp keyword is optional and allows for dropping or passing of SIP 181 messages based on the presence or absence of SDP.
Step 5	**exit** **Example:** `Router(config-dial-peer)# exit`	Exits the current mode.

Configuring Mapping of SIP Provisional Response Messages Globally

Perform this task to configure mapping of specific received SIP provisional response messages at a global level in SIP configuration (conf-serv-sip) mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **sip**
5. **map resp-code 181 to 183**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode, or other security level set by a system administrator. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**voice service voip** **Example:**	Enters voice service VoIP configuration mode.

	Command or Action	Purpose
	`Router(config)# voice service voip`	
Step 4	**sip** **Example:** `Router(conf-voi-serv)# sip`	Enters SIP configuration mode.
Step 5	**map resp-code 181 to 183** **Example:** `Router(conf-serv-sip)# map resp-code 181 to 183`	Enables mapping globally of received SIP messages of a specified message type to a different SIP message type.
Step 6	**exit** **Example:** `Router(conf-serv-sip)# exit`	Exits the current mode.

Configuring Mapping of SIP Provisional Response Messages at the Dial-Peer Level

Perform this task to configure mapping of received SIP provisional response messages at the dial-peer level, in dial peer voice configuration (config-dial-peer) mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag* **voip**
4. **voice-class sip map resp-code 181 to 183**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enters privileged EXEC mode, or other security level set by a system administrator. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**dial-peer voice** *tag* **voip** **Example:** `Router(config)# dial-peer voice 2 voip`	Enters dial peer VoIP configuration mode.
Step 4	**voice-class sip map resp-code 181 to 183** **Example:** `Router(config-dial-peer)# voice-class sip map resp-code 181 to 183`	Enables mapping of received SIP messages of a specified SIP message type on a specific dial peer to a different SIP message type.
Step 5	**exit** **Example:** `Router(config-dial-peer)# exit`	Exits the current mode.

Feature Information for Configuring SIP 181 Call is Being Forwarded Message

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.

Feature History Table entry for the Cisco Unified Border Element.

Table 1: Feature Information for SIP 181 Call is Being Forwarded Messages

Feature Name	Releases	Feature Information
SIP 181 Call is Being Forwarded Message	12.2(13)T	This feature allows users to configure support for SIP 181 Call is Being Forwarded messages either globally or on a specific dial-peer. This feature includes the following new or modified commands: **block**, **map resp-code**, **voice-class sip block, voice-class sip map resp-code**.

Feature History Table entry for the Cisco Unified Border Element (Enterprise).

Table 2: Feature Information for SIP 181 Call is Being Forwarded Messages

Feature Name	Releases	Feature Information
SIP 181 Call is Being Forwarded Message	Cisco IOS XE Release 3.1S	This feature allows users to configure support for SIP 181 Call is Being Forwarded messages either globally or on a specific dial-peer. This feature includes the following new or modified commands: **block**, **map resp-code**, **voice-class sip block, voice-class sip map resp-code**.

CHAPTER 3

Expires Timer Reset on Receiving or Sending SIP 183 Message

This feature enables support for resetting the Expires timer when receiving or sending SIP 183 messages on Cisco Unified Communications Manager Express (Cisco Unified CME), a Cisco IOS voice gateway, or a Cisco Unified Border Element (Cisco UBE). When the terminating device lacks answer supervision or does not send the required SIP 200 OK message within the timer expiry, you can enable this feature to send periodic SIP 183 messages to reset the Expires timer and preserve the call until final response. This feature can be enabled globally or on a specific dial peer. Additionally, you can configure this feature based on the presence or absence of Session Description Protocol (SDP).

For details about enabling this feature, see the **reset timer expires** and **voice-class sip reset timer expires** commands in the Cisco IOS Voice Command Reference.

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.

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

Prerequisites for Expires Timer Reset on Receiving or Sending SIP 183 Message

Before configuring support for Expires timer reset for SIP 183 on Cisco IOS SIP time-division multiplexing (TDM) gateways, Cisco UBEs, or Cisco Unified CME, verify the SIP configuration within the VoIP network for the appropriate originating and terminating gateways as described in the Cisco IOS SIP Configuration Guide.

Cisco Unified Border Element

- Cisco IOS Release 15.0(1)XA or a later release must be installed and running on your Cisco Unified Border Element.

Cisco Unified Border Element (Enterprise)

- Cisco IOS XE Release 3.1S or a later release must be installed and running on your Cisco ASR 1000 Series Router.

How to Configure Expires Timer Reset on Receiving or Sending SIP 183 Message

To configure the Support for Expires Timer Reset on Receiving or Sending SIP 183 Message feature, complete the tasks in this section. You can enable this feature globally, using the **reset timer expires** command in voice service SIP configuration mode, or on a specific dial-peer using the **voice-class sip reset timer expires** command in dial peer voice configuration mode.

Configuring Reset of Expires Timer Globally

Perform this task to enable resetting of the Expires timer at the global level in SIP configuration (conf-serv-sip) mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **sip**
5. **reset timer expires 183**
6. **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:**	Enters global configuration mode.

	Command or Action	Purpose
	`Router# configure terminal`	
Step 3	**voice service voip** **Example:** `Router(config)# voice service voip`	Enters voice service VoIP configuration mode.
Step 4	**sip** **Example:** `Router(conf-voi-serv)# sip`	Enters SIP configuration mode.
Step 5	**reset timer expires 183** **Example:** `Router(conf-serv-sip)# reset timer expires 183`	Enables resetting of the Expires timer upon receipt of SIP 183 messages globally.
Step 6	**exit** **Example:** `Router(conf-serv-sip)# exit`	Exits the current mode.

Configuring Reset of Expires Timer at the Dial-Peer Level

Perform this task to enable resetting of the Expires timer at the dial-peer level in dial peer voice configuration (config-dial-peer) mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag* **voip**
4. **voice-class sip reset timer expires 183**
5. **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.

	Command or Action	Purpose
Step 3	**dial-peer voice** *tag* **voip** **Example:** `Router(config)# dial-peer voice 2 voip`	Enters dial peer VoIP configuration mode.
Step 4	**voice-class sip reset timer expires 183** **Example:** `Router(config-dial-peer)# voice-class sip reset timer expires 183`	Enables resetting of the Expires timer upon receipt of SIP 183 messages on a specific dial peer.
Step 5	**exit** **Example:** `Router(config-dial-peer)# exit`	Exits the current mode.

Feature Information for Configuring Support for Expires Timer Reset on Receiving or Sending SIP 183 Message

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.

Feature History Table entry for the Cisco Unified Border Element.

Table 3: Feature Information for Support for Expires Timer Reset on Receiving or Sending SIP 183 Message

Feature Name	Releases	Feature Information
Support for Expires Timer Reset on Receiving or Sending SIP 183 Message	15.0(1)XA 15.1(1)T	This feature enables support for resetting the Expires timer upon receipt of SIP 183 messages on Cisco Unified Communications Manager Express (Cisco Unified CME), a Cisco IOS voice gateway, or a Cisco Unified Border Element (Cisco UBE). The following commands were introduced or modified: **reset timer expires**and **voice-class sip reset timer expires.**

Feature History Table entry for the Cisco Unified Border Element (Enterprise) .

Table 4: Feature Information for Support for Expires Timer Reset on Receiving or Sending SIP 183 Message

Feature Name	Releases	Feature Information
Support for Expires Timer Reset on Receiving or Sending SIP 183 Message	Cisco IOS XE Release 3.1S	This feature enables support for resetting the Expires timer upon receipt of SIP 183 messages on Cisco Unified Communications Manager Express (Cisco Unified CME), a Cisco IOS voice gateway, or a Cisco Unified Border Element (Cisco UBE). The following commands were introduced or modified: **reset timer expires**and **voice-class sip reset timer expires.**

CHAPTER 4

Cisco Unified Communications Manager Line-Side Support

Note The Cisco Unified Communications Manager (Unified Communications Manager) Lineside feature is no longer supported. The feature is deprecated for Cisco Unified Border Element on Cisco IOS 15.5(2)T Release and later releases. To support this feature, you must configure Cisco Unified Border Element on Cisco IOS 15.4(2)T or prior releases.

Cisco Unified Communications Manager is an enterprise-class IP communications processing system. It extends enterprise telephony features and capabilities to IP phones, media processing devices, VoIP gateways, mobile devices, and multimedia applications. Cisco Unified Border Element (Cisco UBE) provides line-side support for Cisco Unified Communications Manager. This support enables communication between devices (such as phones) used by remote users on different logical networks, in both cloud-based and premise-based deployments.

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.

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

Restrictions for Cisco Unified Communications Manager Line-Side Support

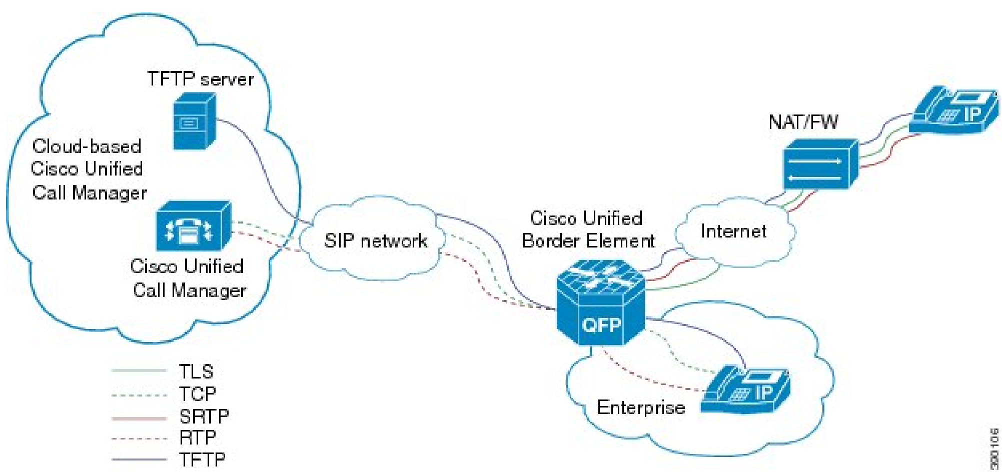
- In Cisco Unified Communications Manager Line-Side Support deployments, Cisco Unified Border Element does not support TFTP encrypted configuration files.

Information About Cisco Unified Communications Manager Line-Side Support

Cisco UBE Line-Side Deployment

In a typical deployment Cisco Unified Border Element (Cisco UBE) is placed between the Cisco Unified Communications Manager and the endpoint. Before invoking a service the phone contacts the CUBE Trivial File Transfer Protocol (TFTP) server to get configuration information such as the Certificate Trust List (CTL) file and phone-specific configuration settings. The phone then registers with Cisco Unified Communications Manager. In the deployment shown below, Cisco Unified Communications Manager and the phone configuration operate in unsecured mode (TCP to Real-Time Transport Protocol). The phone configuration can be changed to operate in a secure mode (Transport Layer Security Secure to Real-Time Transport Protocol) if needed. When the phone registration is completed the phone can invoke all normal call services.

Figure 1: Cisco UBE Line-Side Deployment



Line-Side Support for CUCM on CUBE

For an IP phone to register on a CUCM through CUBE, CUBE must be configured to do the following requirements.

- TCP must be used for registration.

- The MAC address of the device (device ID) and the device name, present in the CONTACT header of the REGISTER message, need to be copied to the outgoing messages and passed to the CUCM intact.

Table 5: Command for Line-Side Support for CUCM on CUBE

Dial-Peer Configuration Mode (config-dial-peer)	Global VoIP Configuration mode (config-voi-serv)
voice-class sip extension cucm	**sip** **extension cucm**

When Line Side Support for CUCM on CUBE feature is configured, the following supported, nonmandatory headers are passed through automatically without the need for further configuration:

- Call-Info
- Content-ID
- Allow-Events
- Supported
- Remote-Party-ID
- Require
- Referred-By

Figure 2: Predefined Supported NonMandatory Headers

```
!-- predefined hidden supported non-mandatory header pass-through list
!-- the list number 20001 is out of user configuration range

voice class sip-hdr-passthrulist 20001
 passthru-hdr Call-Info
 passthru-hdr Content-ID
 passthru-hdr Allow-Events
 passthru-hdr Supported
 passthru-hdr Remote-Party-ID
 passthru-hdr Require
 passthru-hdr Referred-By
```

When Line Side Support for CUCM on CUBE is configured, predefined SIP profiles automatically remove the Cisco-Guide header from the outgoing INVITE.

Figure 3: Predefined SIP Profile

```
!-- predefined hidden sip profile
!-- the profile number 20001 is out of user configuration range

voice class sip-profiles 20001
 request INVITE sip-header Cisco-Guid remove
```

Note If a user explicitly configures the above configurations, ensure that the configurations are merged with the above automatic configurations.

Configuring SIP Extension

You can use the SIP extension to enable support of CUCM-specific features. Configure the SIP extension under dial-peer facing CUCM lineside and CUCM. You can also configure the SIP extension command in global SIP configuration.

SUMMARY STEPS

1. **dial-peer voice** *tag* **voip**
2. **voice-class sip extension** {**cucm** | **system**}
3. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**dial-peer voice** *tag* **voip** **Example:** `Device(config)# dial-peer voice 2 voip`	Enters dial peer configuration mode.
Step 2	**voice-class sip extension** {**cucm** | **system**} **Example:** `Device(config-dial-peer)# voice-class sip extension cucm`	Configures SIP extension to enable support for CUCM. • Use the keyword **system** to configure the SIP extension globally.
Step 3	**end** **Example:** `Device(config-dial-peer)# end`	Returns to privileged EXEC mode.

Configuring a PKI Trustpoint

SUMMARY STEPS

1. **crypto key generate rsa** [**label** *key-label*] [**modulus** *modulus-size*] **general-keys**
2. **crypto pki trustpoint** *name*
3. **enrollment selfsigned**
4. **subject-name** [*x.500-name*]
5. **subject-alt-name** *sip-security-profile-name*
6. **revocation-check** *method1*[*method2* [*method3*]]
7. **rsakeypair** *key-label*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**crypto key generate rsa** [**label** *key-label*] [**modulus** *modulus-size*] **general-keys** **Example:** `Device(config)# crypto key generate rsa label pp_rsa modulus 1024 general-keys`	Generates a RSA key pair. **Note** A self-signed key can only support a *modulus-size* value of 1024 bits.

	Command or Action	Purpose
Step 2	**crypto pki trustpoint** *name* **Example:** `Device(config)# crypto pki trustpoint callmg23`	Declares the trustpoint that the device should use and enters ca-trustpoint configuration mode.
Step 3	**enrollment selfsigned** **Example:** `Device(config-ca-trustpoint)# enrollment selfsigned`	Specifies self-signed enrollment for a trustpoint.
Step 4	**subject-name** [*x.500-name*] **Example:** `Device(config-ca-trustpoint)# subject-name CN=ASR1006-CCN-4`	Specifies the subject name in the certificate request.
Step 5	**subject-alt-name** *sip-security-profile-name* **Example:** `Device(config-ca-trustpoint)# subject-alt-name 6961_SEC.cisco.com 8941_SEC.cisco.com 8945_SEC.cisco.com 7975_SEC.cisco.com 7970_SEC.cisco.com`	Specifies the alternative subject name in the certificate request. • Use the **subject-alt-name** command only when Cisco UBE is interacting with CUCM in secure mode. • The value of **subject-alt-name** must be the SIP security profile name under CUCM.
Step 6	**revocation-check** *method1*[*method2* [*method3*]] **Example:** `Device(config-ca-trustpoint)# revocation-check crl`	Checks the revocation status of a certificate.
Step 7	**rsakeypair** *key-label* **Example:** `Device(config-ca-trustpoint)# rsakeypair pp1`	Specifies which RSA keypair to associate with the certificate.

What to do next

Import the CUCM and CAPF key.

Importing the CUCM and CAPF Key

Before you begin

Download the CUCM key (the CallManager.pem file) from the Cisco Unified Communications Manager Operating System Administration web page.

Login to Cisco Unified OS Administration and Security and Certificate Management, download the CUCM key (the CallManager.pem file), and copy and paste the CUCM key to CUBE

SUMMARY STEPS

1. **crypto pki trustpoint** *name*
2. **revocation-check** *method1*[*method2* [*method3*]]
3. **enrollment terminal**
4. **crypto pki authenticate** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**crypto pki trustpoint** *name* **Example:** `Device(config)# crypto pki trustpoint cucm_trustpoint`	Creates a trustpoint for the CUCM key and enters ca-trustpoint configuration mode.
Step 2	**revocation-check** *method1*[*method2* [*method3*]] **Example:** `Device(config-ca-trustpoint)# revocation-check none`	Checks the revocation status of a certificate.
Step 3	**enrollment terminal** **Example:** `Device(config-ca-trustpoint)# enrollment terminal`	Specifies manual cut-and-paste certificate enrollment.
Step 4	**crypto pki authenticate** *name* **Example:** `Device(config-ca-trustpoint)# crypto pki authenticate cucm_trustpoint`	Authenticates the trustpoint. At the prompt to enter the certificate, copy the contents of the CallManager.pem file that you downloaded above and paste it at the prompt. At the prompt to accept the file, enter “yes”. **Note** When you copy the certificate, ensure that you also copy the BEGIN and END lines.

What to do next

Repeat the above steps for the CAPF key (the CAPF.pem file).

Creating a CTL File

SUMMARY STEPS

1. **voice-ctl-file** *ctl-filename*

2. **record-entry selfsigned trustpoint** *trustpoint-name*
3. **record-entry capf trustpoint** *trustpoint-name*
4. **record-entry cucm-tftp trustpoint** *trustpoint-name*
5. **complete**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**voice-ctl-file** *ctl-filename* **Example:** `Device(config)#voice-ctl-file ct1`	Creates a CTL file and enters CTL file configuration mode.
Step 2	**record-entry selfsigned trustpoint** *trustpoint-name* **Example:** `Device(config-ctl-file)#record-entry selfsigned trustpoint self-trustpoint6s`	Configures the trustpoints to be used for creating the CTL file.
Step 3	**record-entry capf trustpoint** *trustpoint-name* **Example:** `Device(config-ctl-file)#record-entry capf trustpoint capf-trustpoint6s`	Specifies that the trustpoint is created using the CAPF certificate imported from Cisco Unified Communications Manager to the device.
Step 4	**record-entry cucm-tftp trustpoint** *trustpoint-name* **Example:** `Device(config-ctl-file)#record-entry cucm-tftp trustpoint cucm-trustpoint`	Specifies that the trustpoint is created using the specified TFTP and Cisco Unified Communications Manager certificate imported to the device.
Step 5	**complete** **Example:** `Device(config-ctl-file)# complete`	Completes the CTL-file creation.

Configuring a Phone Proxy

SUMMARY STEPS

1. **voice-phone-proxy** *phone-proxy-name*
2. **voice-phone-proxy file-buffer** *size*
3. **tftp-server-address** [**ipv4** *server-ip-address* | *domain-name*]
4. **ctl-file** *ctl-filename*
5. **access-secure**
6. **complete**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**voice-phone-proxy** *phone-proxy-name* **Example:** `Device(config)# voice-phone-proxy pp`	Configures a phone proxy and enters phone-proxy configuration mode.
Step 2	**voice-phone-proxy file-buffer** *size* **Example:** `Device(config)# voice-phone-proxy file-buffer 30`	Configures the phone-proxy file buffering parameter, in MB.
Step 3	**tftp-server-address** [**ipv4** *server-ip-address* | *domain-name*] **Example:** `Device(config-phone-proxy)# tftp-server-address ipv4 172.110.36.2`	Configures the TFTP server address.
Step 4	**ctl-file** *ctl-filename* **Example:** `Device(config-phone-proxy)# ctl-file ct1`	Configures the CTL filename.
Step 5	**access-secure** **Example:** `Device(config-phone-proxy)# access-secure`	Specifies that the secure (encrypted) mode is to be used for access.
Step 6	**complete** **Example:** `Device(config-phone-proxy)# complete`	Completes the phone-proxy configuration.

Attaching a Phone Proxy to a Dial Peer

SUMMARY STEPS

1. **dial-peer voice** *tag* **voip**
2. **phone-proxy** *phone-proxy-name* **signal-addr ipv4** *ipv4-address* **cucm ipv4** *ipv4-address*
3. **session protocol sipv2**
4. **session target registrar**
5. **session transport** {**udp** | **tcp** [tls]}
6. **incoming uri** {**from** | **request** | **to** | **via**} *tag*
7. **destination uri** *tag*
8. **voice-class sip call-route url**

9. **voice-class sip profiles** *number*
10. **voice-class sip registration passthrough** [**registrar-index** *index*]
11. **voice-class sip pass-thru headers**
12. **voice-class sip copy-list** {*tag* | **system**}
13. **codec transparent**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**dial-peer voice** *tag* **voip** **Example:** `Device(config)# dial-peer voice 10 voip`	Defines a particular dial peer, specifies the method of voice encapsulation, and enters dial peer configuration mode.
Step 2	**phone-proxy** *phone-proxy-name* **signal-addr ipv4** *ipv4-address* **cucm ipv4** *ipv4-address* **Example:** `Device(config-dial-peer)# phone-proxy pp1 signal-addr ipv4 10.0.0.8 cucm ipv4 198.51.100.1`	Configures the phone proxy for the related dial peer.
Step 3	**session protocol sipv2** **Example:** `Device(config-dial-peer)# session protocol sipv2`	Specifies a session protocol (SIPv2) for calls between local and remote devices.
Step 4	**session target registrar** **Example:** `Device(config-dial-peer)# session target registrar`	Specifies that a call from a VoIP dial peer is routed to the registrar end point.
Step 5	**session transport** {**udp** | **tcp** [tls]} **Example:** `Device(config-dial-peer)# session transport tcp tls`	Configures the underlying transport layer protocol for SIP messages to transport layer security over TCP (TLS over TCP).
Step 6	**incoming uri** {**from** | **request** | **to** | **via**} *tag* **Example:** `Device(config-dial-peer)# incoming uri request 11`	Specifies the voice class used to match the VoIP dial peer to the uniform resource identifier (URI) of an incoming call. Any request matching “uri 11” is destined to this dial peer.
Step 7	**destination uri** *tag* **Example:** `Device(config-dial-peer)# destination uri 12`	Specifies the voice class used to match a dial peer to the destination URI of an outgoing call. Any request matching “uri 12” is destined to this dial peer.

	Command or Action	Purpose
Step 8	**voice-class sip call-route url** **Example:** `Device(config-dial-peer)# voice-class sip call-route url`	Enables call routing based on the URL.
Step 9	**voice-class sip profiles** *number* **Example:** `Device(config-dial-peer)# voice-class sip profiles 10`	Configures a SIP profile for a voice class.
Step 10	**voice-class sip registration passthrough** [**registrar-index** *index*] **Example:** `Device(config-dial-peer)# voice-class sip registration passthrough registrar-index 1`	Configures the SIP registration pass-through options on the dial peer.
Step 11	**voice-class sip pass-thru headers** **Example:** `Device(config-dial-peer)# voice-class sip pass-thru headers 10`	Configures a list of headers for pass through by referring to a globally configured list.
Step 12	**voice-class sip copy-list** {*tag* | **system**} **Example:** `Device(config-dial-peer)# voice-class sip copy-list 10`	Configures the list of entities to be sent to the peer call leg.
Step 13	**codec transparent** **Example:** `Device(config-dial-peer)# codec transparent`	Enables codec capabilities to be passed transparently between endpoints in a Cisco Unified Border Element.

Verifying CUCM Lineside Support

The **show** commands can be entered in any order.

SUMMARY STEPS

1. **enable**
2. **show dial-peer voice** *dial-peer-id* | **section voice class sip extension**
3. **show dial-peer voice**
4. **show voice class phone-proxy**
5. **show voice class phone-proxy sessions**

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode.

- Enter your password if prompted.

Example:

```
Device> enable
```

Step 2 **show dial-peer voice** *dial-peer-id* | **section voice class sip extension**

Example:

```
CUBE# show dial-peer voice 5678 | section voice class sip extension

voice class sip extension = system,
```

Displays if **extension cucm** has not been configured for the dial peer.

Example:

```
CUBE# show dial-peer voice 5678 | section voice class sip extension

voice class sip extension = cucm,
```

Displays if **extension cucm** has been configured for the dial peer.

Example:

```
CUBE# show dial-peer voice 5678 | section voice class sip extension

voice class sip extension = none,
```

Displays if **extension cucm** has been removed for the dial peer using the **no** form of the command.

Step 3 **show dial-peer voice**

Example:

```
Device# show dial-peer voice 100

voice class sip extension = system,
voice class sip contact-passing = system,
voice class sip requri-passing = system,
voice class phone proxy name: phone_proxy_secure
voice class phone proxy config: complete
```

Step 4 **show voice class phone-proxy**

Example:

```
Device# show voice class phone-proxy

Phone-Proxy 'phone_proxy':
Description:
 Access Secure: non-secure (default)
Tftp-server address: 20.21.27.146
Capf server address: 20.21.27.146
CUCM service settings: preserve (default)
CTL file name: ctl_file
```

```
Session-timeout: 180 seconds
Max-concurrent-sessions: 30
Current sessions: 0
TFTP sessions: 0
HTTP download sessions: 0
HTTP application sessions: 0
CAPF sessions: 0
Config status: complete
SIP dial-peers associated:
   Name
   ---------------
   1
--------------------------------------------------------------

Phone-Proxy 'phone_proxy_secure':
Description:
 Access Secure: secure
Tftp-server address: 20.21.27.146
Capf server address: 20.21.27.146
CUCM service settings: preserve (default)
CTL file name: ctl_file
Session-timeout: 180 seconds
Max-concurrent-sessions: 30
Current sessions: 0
TFTP sessions: 0
HTTP download sessions: 0
HTTP application sessions: 0
CAPF sessions: 0
Config status: complete
SIP dial-peers associated:
   Name
   ---------------
   3
   dialpeer4
--------------------------------------------------------------
```

Step 5 **show voice class phone-proxy sessions**

Example:

```
Device# show voice class phone-proxy sessions

Phone-Proxy 'phone_proxy_ipad':
                    Source                              Destination
 ------------------------------ Sessions of Dial-peer 5             -------------------------------
|Access: 10.74.9.219         :45232         10.74.9.209         :6970
  |
|Core  :   20.21.29.209       :45300          20.21.27.146      :6970
 |
----------------------------------------------------------------------------------------------------
```

Example: Configuring a PKI Trustpoint

```
Device(config)# crypto key generate rsa label pp_rsa modulus 1024 general-keys
Device(config)# crypto pki trustpoint callmg23
Device(config-ca-trustpoint)# enrollment selfsigned
```

```
Device(config-ca-trustpoint)# subject-name CN=ASR1006-CCN-4
Device(config-ca-trustpoint)# subject-alt-name 6961_SEC.cisco.com 8941_SEC.cisco.com
8945_SEC.cisco.com 7975_SEC.cisco.com 7970_SEC.cisco.com
Device(config-ca-trustpoint)# revocation-check crl
Device(config-ca-trustpoint)# rsakeypair pp1
```

Example: Importing the CUCM and CAPF Key

The following example shows how to import the CUCM and CAPF key after you have downloaded the CUCM key (the CallManager.pem file) and the CAPF key (the CAPF.pem file) from the Cisco Unified Communications Manager Operating System Administration web page.

```
Device(config)# crypto pki trustpoint cucm_trustpoint
 Device(config-ca-trustpoint)# revocation-check none
 Device(config-ca-trustpoint)# enrollment terminal
 Device(config-ca-trustpoint)# crypto pki authenticate cucm_trustpoint
```

Example: Creating a CTL File

```
Device(config)# voice-ctl-file ct1
Device(config-ctl-file)# record-entry selfsigned trustpoint self-trustpoint6s
Device(config-ctl-file)# record-entry capf trustpoint capf-trustpoint6s
Device(config-ctl-file)# record-entry cucm-tftp trustpoint cucm-trustpoint
Device(config-ctl-file)# complete
```

Example: Configuring a Phone Proxy

```
Device(config)# voice-phone-proxy pp
Device(config-phone-proxy)# voice-phone-proxy pp
Device(config-phone-proxy)# voice-phone-proxy file-buffer size 30
Device(config-phone-proxy)# tftp-server address ipv4 172.110.36.2
Device(config-phone-proxy)# ctl-file ct1
Device(config-phone-proxy)# access-secure
Device(config-phone-proxy)# complete
```

Example: Attaching a Phone Proxy to a Dial Peer

```
Device(config)# dial-peer voice 10 voip
 Device(config-dial-peer)# phone-proxy pp1 signal-addr ipv4 10.0.0.8 cucm ipv4 198.51.100.1

 Device(config-dial-peer)# session-protocol sipv2
 Device(config-dial-peer)# session target registrar
 Device(config-dial-peer)# session transport tcp tls
 Device(config-dial-peer)# incoming uri request 11
 Device(config-dial-peer)# destination uri 12
```

```
Device(config-dial-peer)# voice-class sip call-route url
Device(config-dial-peer)# voice-class sip profiles 10
Device(config-dial-peer)# voice-class sip registration passthrough registrar-index 1
Device(config-dial-peer)# voice-class sip passthrough headers 10
Device(config-dial-peer)# voice-class sip copy-list 10
Device(config-dial-peer)# codec transparent
```

Feature Information for Cisco Unified Communications Manager Line-Side Support

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 Cisco Unified Communications Manager Line-Side Support

Feature Name	Releases	Feature Information
Cisco Unified Communications Manager Line-Side Support	15.5(2)T	The Cisco Unified Communications Manager (CUCM) Line-Side Support feature was supported until the release 15.4(2)T. This feature has been deprecated from 15.5(2)T release onwards.
Simplified Line-Side Support of CUCM on CUBE	15.4(2)T Cisco IOS XE Release 3.12S	The Simplified Line-Side Support of CUCM on CUBE feature simplifies the complex CUBE configurations required for registering IP Phones on a CUCM through CUBE using a single CLI that automatically applies all the necessary configurations. The following commands were modified by this feature: **extension cucm** and **voice-class sip extension cucm**.

Feature Name	Releases	Feature Information
Cisco Unified Communications Manager Line-Side Support	15.3(3)M Cisco IOS XE Release 3.10S	The Cisco Unified Communications Manager Line-Side Support feature provides line-side support for Cisco Unified Communications Manager and IP phones deployed on different logical networks, in both cloud-based and premise-based deployments. The following commands were introduced or modified: **access-secure**, **capf-address**, **clear voice phone-proxy all-sessions**, **complete (ctl file)**, **ctl-file (phone proxy)**, **debug voice phone-proxy**, **description (ctl file)**, **description (phone proxy)**, **disable service-settings**, **max-concurrent-sessions**, **phone-proxy (dial peer)**, **port-range**, **record-entry**, **show voice class ctl-file**, **show voice class phone-proxy**, **service-map**, **session-timeout**, **tftp-server address**, **voice-ctl-file**, **voice-phone-proxy**.

CHAPTER 5

Cisco Unified Border Element Intercluster Lookup Service

The Cisco Unified Border Element (CUBE) Intercluster Lookup Service feature enables Cisco Unified Communications Manager to establish calls using Uniform Resource Identifiers (URIs.) It provides a framework for sharing information about user-contact information between Cisco Unified Communications Manager clusters. All URIs being used within a cluster are grouped together and associated with a cluster identifier called a route string. To interoperate with Cisco Unified Communications Manager, CUBE is enhanced to route the call based on the received destination route string. This feature works with Cisco Unified Communication Manager Version 9.5 and later.

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.

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

Information About CUBE Intercluster Lookup Service

CUBE Intercluster Lookup Service Overview

A Uniform Resource Identifier (URI) is a device-independent user address. A subscriber can use a URI as a personal identity and move from one network to another without any change in the URI. You cannot summarize URIs within an enterprise network (for example, abc@company.com) the same way that directory number ranges are summarized.

The Intercluster Lookup Services is a dynamic mechanism to discover URIs. When it is enabled, Cisco Unified Communications Manager users can initiate calls using URIs. The Intercluster Loookup Service provides a framework for sharing user-contact information between Cisco Unified Communications Manager clusters. All URIs being used within a cluster are grouped together and associated with a cluster identifier called a route string. These URI groups and their associated route strings are shared between all other participating clusters.

While initiating a call, the URI uses the Intercluster Lookup Service to identify the target URI and associated route string to route the call between clusters. Cisco Unified Communications Manager uses a Session Initiation Protocol (SIP) route pattern to match the route string returned by Intercluster Lookup Service and route the call over a SIP trunk. If Intercluster Lookup Service is enabled, the Cisco Unified Communications Manager SIP trunk sends the SIP invite message with destination route string header information.

To interoperate with Cisco Unified Communications Manager, CUBE is enhanced to route the call based on the received destination route string. CUBE supports exact match and wildcard match for a route string and parses the received destination route string header and routes a call forward to the destination. The destination can be a Cisco Unified Communications Manager cluster, public switched telephone network (PSTN), or any third-party unified communications device.

The dial-peer module is enhanced to support the dial-peer matching based on the destination route string header. The destination route string is used to match an outbound dial peer. The match can be an exact match or wildcard match.

For example, consider London.UK.EU as the route string. The SIP dial-peer configuration is as follows:

- Dial-peer 1: London.UK.EU
- Dial-peer 2: *.UK.EU
- Dial-peer 3: *.EU

The destination route string header and route string match are not case-sensitive. In this scenario, London.UK.EU and london.uk.eu match dial-peer 1 and therefore, dial-peer 1 is selected for outbound process.

If call routing policies are enabled, call routing based on a destination route string takes precedence over any other routing configurations. For example, if call routing is configured on a destination route string globally or at the dial-peer level, the call is routed considering the destination route string. If no match is found, then the call is routed using other URLs and header configuration options.

CUBE Support for URIs

For URI dialing from the Cisco Unified Communications Manager phone, use the URI in user@dest-route-string format. By default, CUBE supports only numeric E164 numbers in the user-part of the request line and headers (For example, +123456789@dest-route-string). As an administrator, you can leverage the CUBE feature Domain-Based Routing’s **call-route url** command by enabling support for the alphanumeric user-part in the request line. Without this command, an alphanumeric URI fails call routing on CUBE with a 484 Address Incomplete error.

For more information on Domain-Based Routing feature, see https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/voice/cube/configuration/cube-book/voi-domainbased-routing.html.

Similarly, the URI-Based Dialing Enhancements feature includes support for call routing on CUBE when the user-part of the incoming request URI is non-E164. By default, the CUBE converts the @dest-route-string format of the request URI to the session target IP address of the outbound dial-peer. You can configure CUBE to pass through the full SIP URI (@dest-route-string) from the inbound call-leg without modification by using the URI-Based Dialing Enhancement's **requri-passing** command. In addition, you can use URI information to route calls using the **session target sip-uri** command.

For more information on URI-Based Dialing Enhancements feature, see https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/voice/cube/configuration/cube-book/sip-supp-uri-based-dialing.html.

How to Configure CUBE Intercluster Lookup Service

Configuring a Route String Pattern

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice class route-string** *tag*
4. **pattern** *string*
5. **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 class route-string** *tag* **Example:** `Device(config)# voice class route-string 2`	Enters voice class configuration mode.
Step 4	**pattern** *string* **Example:** `Device(config-class)# pattern london.uk.eu`	Configures a pattern string in the specified route string. **Note** Multiple patterns can be configured under one route string class and the same route string class can be configured under multiple dial-peers. You also can use an asterisk (*) as the wildcard match option while provisioning the pattern.
Step 5	**end** **Example:**	Exits voice class configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-class)# end`	

Configuring a Call Route on a Destination Route String Globally

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **sip**
5. **call-route dest-route-string**
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 voice service configuration mode.
Step 4	**sip** **Example:** `Device(conf-voi-serv)# sip`	Enters SIP configuration mode.
Step 5	**call-route dest-route-string** **Example:** `Device(conf-serv-sip)# call-route dest-route-string`	Configures call routing globally on a destination route string. **Note** By default, call routing on a destination route string is disabled.
Step 6	**end** **Example:**	Exits SIP configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(conf-serv-sip)# end`	

Configuring a Route String Passthrough List Header

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice class sip-hdr-passthrulist** *tag*
4. **passthru-hdr** *name*
5. **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 class sip-hdr-passthrulist** *tag* **Example:** `Device(config)# voice class sip-hdr-passthrulist 2`	Enters voice class configuration mode.
Step 4	**passthru-hdr** *name* **Example:** `Device(config-class)# passthru-hdr x-cisco-dest-route-string`	Configures header to be added to the route string passthrough list.
Step 5	**end** **Example:** `Device(config-class)# end`	Exits voice class configuration mode and returns to privileged EXEC mode.

Configuring a Destination Route String Call Route at the Dial-Peer Level

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag* **voip**
4. **description** *string*
5. **destination route-string** *tag*
6. **session protocol sipv2**
7. **session target ipv4:***destination address*
8. **voice-class sip call-route dest-route-string**
9. **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	**dial-peer voice** *tag* **voip** **Example:** `Device(config)# dial-peer voice 1 voip`	Enters dial peer voice configuration mode.
Step 4	**description** *string* **Example:** `Device(config-dial-peer)# description outbound-dialpeer`	Adds descriptive information about the dial peer.
Step 5	**destination route-string** *tag* **Example:** `Device(config-dial-peer)# destination route-string 2`	Configures a destination route string for the dial peer. **Note** By default, the call route on a destination route string is disabled. The destination route string call route configuration at the dial-peer level takes precedence over the global configuration when routing a call.
Step 6	**session protocol sipv2** **Example:**	Configures the IETF Session Initiation Protocol (SIP) for the dial peer.

	Command or Action	Purpose
	`Device(config-dial-peer)# session protocol sipv2`	
Step 7	**session target ipv4:***destination address* **Example:** `Device(config-dial-peer)# session target ipv4:192.0.2.6`	Configures the session target IP address of the dial peer.
Step 8	**voice-class sip call-route dest-route-string** **Example:** `Device(config-dial-peer)# voice-class sip call-route dest-route-string`	Configures call routing on the destination route string for a dial peer.
Step 9	**end** **Example:** `Device(config-dial-peer)# end`	Exits dial peer voice configuration mode and returns to privileged EXEC mode.

Configuring a Route String Header Pass-Through Using Pass-Through List

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice class sip-hdr-passthrulist** *list-tag*
4. **passthru-hdr** *header-name*
5. **passthru-hdr-unsupp**
6. **exit**
7. **dial-peer voice** *tag* **voip**
8. **description** *string*
9. **session protocol sipv2**
10. **voice-class sip pass-thru headers** *list-tag*
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**voice class sip-hdr-passthrulist** *list-tag* **Example:** `Device(config)# voice class sip-hdr-passthrulist 101`	Configures list of headers to be passed through and enters voice class configuration mode.
Step 4	**passthru-hdr** *header-name* **Example:** `Device(config-class)# passthru-hdr Resource-Priority`	Adds header name to the list of headers to be passed through. Repeat this step for every non-mandatory header.
Step 5	**passthru-hdr-unsupp** **Example:** `Device(config-class)# passthru-hdr-unsupp`	Adds the unsupported headers to the list of headers to be passed through.
Step 6	**exit** **Example:** `Device(config-class)# exit`	Exits the current configuration session and returns to global configuration mode.
Step 7	**dial-peer voice** *tag* **voip** **Example:** `Device(config)# dial-peer voice 1 voip`	Enters dial peer voice configuration mode.
Step 8	**description** *string* **Example:** `Device(config-dial-peer)# description inbound-dialpeer`	Adds descriptive information about the dial peer.
Step 9	**session protocol sipv2** **Example:** `Device(config-dial-peer)# session protocol sipv2`	Configures the IETF Session Initiation Protocol (SIP) for the dial peer.
Step 10	**voice-class sip pass-thru headers** *list-tag* **Example:** `Device(config-dial-peer)# voice-class sip pass-thru headers 101`	Enables call routing based on the destination route string for a dial peer.
Step 11	**end** **Example:**	Exits the current configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-dial-peer)# end`	

Verifying CUBE Intercluster Lookup Service Configuration

The **show** commands can be entered in any order.

SUMMARY STEPS

1. **enable**
2. **show voice class route-string**
3. **show call active voice**
4. **show call history voice**
5. **show sip call**

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode.

- Enter your password if prompted.

Example:

```
Device> enable
```

Step 2 **show voice class route-string**

Displays the call route-string status for voice ports.

Example:

```
Device# show voice class route-string
voice class route-string 2:
  pattern london.uk.eu
  configured in dial-peers: 7 4 6
```

Step 3 **show call active voice**

Displays call information for voice calls in progress. The sample output below shows the destination route string configuration.

Example:

```
Device# show call active voice
DestinationRouteStr=london.uk.eu
```

Step 4 **show call history voice**

Displays the call history table for voice calls. The sample output below shows the destination route string configuration.

Example:

```
Device# show call history voice | in Des
DestinationRouteStr=london.uk.eu
```

Step 5 **show sip call**

Displays active user agent client (UAC) and user agent server (UAS) information on SIP calls.

Example:

```
Device# show sip call
Total SIP call legs:2, User Agent Client:1, User Agent Server:1
SIP UAC CALL INFO
Call 1
SIP Call ID                : 5A4CAE55-E48D11E2-802BDD60-8693A1D1@192.0.2.1
   State of the call       : STATE_ACTIVE (7)
   Substate of the call    : SUBSTATE_NONE (0)
   Calling Number          : 345111
   Called Number           :
   Bit Flags               : 0xC04018 0x10000100 0x80
   CC Call ID              : 12
   Source IP Address (Sig ): 192.0.2.1
   Destn SIP Req Addr:Port : [192.0.2.6]:5060
   Destn SIP Resp Addr:Port: [192.0.2.6]:5060
   Destination Name        : 192.0.2.6
   Number of Media Streams : 1
   Number of Active Streams: 1
   RTP Fork Object         : 0x0
   Media Mode              : flow-through
   Media Stream 1
     State of the stream      : STREAM_ACTIVE
     Stream Call ID           : 12
     Stream Type              : voice-only (0)
     Stream Media Addr Type   : 1
     Negotiated Codec         : g711ulaw (160 bytes)
     Codec Payload Type       : 0
     Negotiated Dtmf-relay    : inband-voice
     Dtmf-relay Payload Type  : 0
     QoS ID                   : -1
     Local QoS Strength       : BestEffort
     Negotiated QoS Strength  : BestEffort
     Negotiated QoS Direction : None
     Local QoS Status         : None
     Media Source IP Addr:Port: [192.0.2.1]:16406
     Media Dest IP Addr:Port  : [192.0.2.6]:6020


Options-Ping    ENABLED:NO    ACTIVE:NO
   Number of SIP User Agent Client(UAC) calls: 1

SIP UAS CALL INFO
Call 1
SIP Call ID                : 1-27273@192.0.2.6
   State of the call       : STATE_ACTIVE (7)
   Substate of the call    : SUBSTATE_NONE (0)
   Calling Number          : 345111
   Called Number           : alice
   Bit Flags               : 0xC0401C 0x10000100 0x4
   CC Call ID              : 11
   Source IP Address (Sig ): 192.0.2.1
   Destn SIP Req Addr:Port : [192.0.2.6]:5061
   Destn SIP Resp Addr:Port: [192.0.2.6]:5061
   Destination Name        : 192.0.2.6
   Destination Route String: london.uk.eu  //This is the configured dest-route-string pattern.//
```

```
   Number of Media Streams : 1
   Number of Active Streams: 1
   RTP Fork Object         : 0x0
   Media Mode              : flow-through
   Media Stream 1
     State of the stream      : STREAM_ACTIVE
     Stream Call ID           : 11
     Stream Type              : voice-only (0)
     Stream Media Addr Type   : 1
     Negotiated Codec         : g711ulaw (160 bytes)
     Codec Payload Type       : 0
     Negotiated Dtmf-relay    : inband-voice
     Dtmf-relay Payload Type  : 0
     QoS ID                   : -1
     Local QoS Strength       : BestEffort
     Negotiated QoS Strength  : BestEffort
     Negotiated QoS Direction : None
     Local QoS Status         : None
     Media Source IP Addr:Port: [192.0.2.1]:16404
     Media Dest IP Addr:Port  : [192.0.2.6]:6000


Options-Ping    ENABLED:NO    ACTIVE:NO
   Number of SIP User Agent Server(UAS) calls: 1
```

Configuration Examples for CUBE Intercluster Lookup Service

Example: Configuring a Route String Pattern

```
Device> enable
Device# configure terminal
Device(config)# voice class route-string 2
Device(config-class)# pattern london.uk.eu
Device(config-class)# pattern *.uk.eu
Device(config-class)# pattern *.eu
Device(config-class)# end
```

Example: Configuring a Call Route on a Destination Route String Globally

```
Device> enable
Device# configure terminal
Device(config)# voice service voip
Device(conf-voi-serv)# sip
Device(conf-serv-sip)# call-route dest-route-string
Device(conf-serv-sip)# end
```

Example: Configuring a Route String Passthrough List Header

```
Device> enable
Device# configure terminal
```

```
Device(config)# voice class sip-hdr-passthrulist 2
Device(config-class)# passthru-hdr x-cisco-dest-route-string
```

Example: Configuring a Destination Route String Call Route at the Dial-Peer Level

```
Device> enable
Device# configure terminal
Device# dial-peer voice 1 voip
Device(config-dial-peer)# description outbound-dialpeer
Device(config-dial-peer)# destination route-string 2
Device(config-dial-peer)# session protocol sipv2
Device(config-dial-peer)# session target ipv4:192.0.2.6
Device(config-dial-peer)# voice-class sip call-route dest-route-string
```

Example: Configuring a Route String Header Pass-Through Using Pass-Through List

```
Device> enable
Device# configure terminal
Device(config)# voice class sip-hdr-passthrulist 101
Device(config-class)# passthru-hdr X-hdr-1
Device(config-class)# passthru-hdr Resource-Priority
Device(config-class)# passthru-hdr-unsupp
Device(config-class)# exit
Device(config)# dial-peer voice 1 voip
Device(config-dial-peer)# description inbound-dialpeer
Device(config-dial-peer)# session protocol sipv2
Device(config-dial-peer)# voice-class sip pass-thru headers 101
Device(config-dial-peer)# end
```

Feature Information for CUBE Intercluster Lookup Service

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 CUBE Intercluster Look Up Service

Feature Name	Releases	Feature Information
CUBE Intercluster Lookup Service (ILS)	15.3(3)M	The CUBE Intercluster Lookup Service feature enables Cisco Unified Communications Manager to establish calls using Uniform Resource Identifiers (URIs.) It provides a framework for sharing information about user-contact information between Cisco Unified Communications Manager clusters. All URIs being used within a cluster are grouped and associated with a cluster identifier called a route string. To interoperate with Cisco Unified Communications Manager, CUBE is enhanced to route the call based on the received destination route string. This feature works with Cisco Unified Communication Manager Version 9.5 and later. The following commands were introduced or modified: **call-route**,**destination route-string**,**passthru-hdr**,**voice class route-string**,**voice class sip-hdr-passthrulist**,**voice-class sip call-route**,**show call active voice**,**show call history voice**.

Feature Name	Releases	Feature Information
CUBE Intercluster Lookup Service (ILS)	Cisco IOS XE Release 3.10S	The CUBE Intercluster Lookup Service feature enables Cisco Unified Communications Manager to establish calls using Uniform Resource Identifiers (URIs.) It provides a framework for sharing information about user-contact information between Cisco Unified Communications Manager clusters. All URIs being used within a cluster are grouped and associated with a cluster identifier called a route string. To interoperate with Cisco Unified Communications Manager, CUBE is enhanced to route the call based on the received destination route string. This feature works with Cisco Unified Communication Manager Version 9.5 and later. The following commands were introduced or modified: **call-route**,**destination route-string**,**passthru-hdr**,**voice class route-string**,**voice class sip-hdr-passthrulist**,**voice-class sip call-route**,**show call active voice**,**show call history voice**.

CHAPTER 6

Additional References

The following sections provide references related to the CUBE Configuration Guide.

Related References

Related Topic	Document Title
Feature Navigator	For information about platforms supported, and Cisco IOS software image support., search by Featur Name listed in Feature Information Table in www.cisco.com/go/cfn
Bug Search Tool Kit	For information about latest caveats and feature information, see Bug Search Tool
Cisco IOS commands	Cisco IOS 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 troubleshootin information--at http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/voice/config_library/15-mt/cube-15-mt-library.htm
Related Application Guides	• *Cisco Unified Communications Manager and Cisco IOS Interoperability Guide* • *Cisco IOS SIP Configuration Guide* • Cisco Unified Communications Manager (CallManager) Programming Guides

Related Topic	Document Title
Troubleshooting and Debugging guides	• Cisco IOS Debug Command Reference, Release 15.3. • *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	Title
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 2806	*URLs for Telephone Calls*
RFC 2833	*RTP Payload for DTMF Digits, Telephony Tones and Telephony Signals*
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.