# CONTENTS

## CHAPTER 1

**Cisco Unified Border Element Protocol-Independent Features and Setup** 1

- Finding Feature Information 1
- Cisco Unified Border Element Protocol-Independent Features and Setup 1
- Toll Fraud Prevention 4

## CHAPTER 2

**Interworking Between RSVP Capable and RSVP Incapable Networks** 7

- Finding Feature Information 7
- Prerequisites for Interworking Between RSVP Capable and RSVP Incapable Networks 8
- Restrictions for Interworking Between RSVP Capable and RSVP Incapable Networks 8
- How to Configure Interworking Between RSVP Capable and RSVP Incapable Networks 8
  - Configuring RSVP on an Interface 8
  - Configuring Optional RSVP on the Dial Peer 9
  - Configuring Mandatory RSVP on the Dial Peer 11
  - Configuring Midcall RSVP Failure Policies 12
  - Configuring DSCP Values 14
  - Configuring an Application ID 15
  - Configuring Priority 16
- Troubleshooting for Interworking Between RSVP Capable and RSVP Incapable Networks Feature 17
- Verifying Interworking Between RSVP Capable and RSVP Incapable Networks Feature Information for Interworking Between RSVP Capable and RSVP Incapable Networks 18

## CHAPTER 3

**SIP INFO Method for DTMF Tone Generation** 21

- Finding Feature Information 21
- Prerequisites for SIP INFO Method for DTMF Tone Generation 21
- Restrictions for SIP INFO Methods for DTMF Tone Generation 22
- Information About SIP INFO Method for DTMF Tone Generation 22
- How to Review SIP INFO Messages 22
### CHAPTER 4

**WebEx Telepresence Media Support Over Single SIP Session**

- Finding Feature Information 27
- Restrictions for WebEx Telepresence Media Support Over Single SIP Session 27
- Information About WebEx Telepresence Media Support Over Single SIP Session 28
- Monitoring WebEx Telepresence Media Support Over Single SIP Session 28
- Feature Information for WebEx Telepresence Media Support Over Single SIP Session 31

### CHAPTER 5

**DTMF Events through SIP Signaling**

- Finding Feature Information 33
- Prerequisites for DTMF Events through SIP Signaling 34
- Restrictions for DTMF Events through SIP Signaling 34
- DTMF Dialing 34
- NOTIFY Messages 34
- Configuring DTMF Events through SIP Signaling 35
  - Verifying SIP DTMF Support 36
- Troubleshooting Tips 41
- Feature Information for DTMF Events through SIP Signaling 41

### CHAPTER 6

**Call Progress Analysis Over IP-to-IP Media Session**

- Feature Information for Call Progress Analysis Over IP-IP Media Session 45
- Restrictions for Call Progress Analysis Over IP-to-IP Media Session 46
- Information About Call Progress Analysis Over IP-IP Media Session 47
  - Call Progress Analysis 47
  - CPA Events 47
- How to Configure Call Progress Analysis Over IP-to-IP Media Session 48
  - Enabling CPA and Setting the CPA Parameters 48
  - Verifying the Call Progress Analysis Over IP-to-IP Media Session 50
  - Troubleshooting Tips 51
- Configuration Examples for the Call Progress Analysis Over IP-to-IP Media Session 51
  - Example: Enabling CPA and Setting the CPA Parameters 51
CHAPTER 7  Codec Preference Lists  53

Feature Information for Negotiation of an Audio Codec from a List of Codecs  53
Codecs configured using Preference Lists  54
Prerequisites for Codec Preference Lists  55
Restrictions for Codecs Preference Lists  55
How to Configure Codec Preference Lists  56

Configuring Audio Codecs Using a Codec Voice Class and Preference Lists  56
Disabling Codec Filtering  57
Troubleshooting Negotiation of an Audio Codec from a List of Codecs  59
Verifying Negotiation of an Audio Codec from a List of Codecs  59

CHAPTER 8  AAC-LD MP4A-LATM Codec Support on Cisco UBE  63

Finding Feature Information  63
Restrictions for AAC-LD MP4A-LATM Codec Support on Cisco UBE  64
AAC-LD MP4A-LATM Codec Support on Cisco UBE  64
How to Configure the MP4A-LATM Codec  65

Configuring the MP4A-LATM Codec on a Dial Peer  65
Configuring the MP4A-LATM Codec under Voice Class Codec  67
Verifying an Audio Call  68

Configuration Examples for AAC-LD MP4A-LATM Codec Support on Cisco UBE  69

Example: Configuring the MP4A-LATM Codec under a Dial Peer  69
Example: Configuring the MP4A-LATM Codec under Voice Class Codec  69
Feature Information for AAC-LD MP4A-LATM Codec Support on Cisco UBE  70

CHAPTER 9  Multicast Music-on-Hold Support on Cisco UBE  71

Prerequisites for Multicast Music-on-Hold Support on Cisco UBE  71
Restrictions for Multicast Music-on-Hold Support on Cisco UBE  71
Information About Multicast Music-on-Hold Support on Cisco UBE  72

Multicast Music-on-Hold  72
How to Enable Multicast Music-on-Hold on Cisco UBE  72

Enabling MMOH on Cisco UBE  72
Verifying the MMOH Support on Cisco UBE  74
Troubleshooting Tips  76

Configuration Examples for Multicast Music-on-Hold Support on Cisco UBE  77
### Network-Based Recording

- Feature Information for Network-Based Recording
- Restrictions for Network-Based Recording
- Information About Network-Based Recording Using CUBE
  - Deployment Scenarios for CUBE-based Recording
- Open Recording Architecture
  - Network Layer
  - Capture and Media Processing Layer
  - Application Layer
- Media Forking Topologies
  - Media Forking with Cisco UCM
  - Media Forking without Cisco UCM
- SIP Recorder Interface
- Metadata
- How to Configure Network-Based Recording
  - Configuring Network-Based Recording (with Media Profile Recorder)
  - Configuring Network-Based Recording (without Media Profile Recorder)
- Verifying the Network-Based Recording Using CUBE
- Additional References for Network-Based Recording

### Video Recording - Additional Configurations

- Feature Information for Video Recording - Additional Configurations
- Information About Additional Configurations for Video Recording
  - Full Intra-Frame Request
- How to Configure Additional Configurations for Video Recording
  - Enabling FIR for Video Calls (Using RTCP of SIP INFO)
  - Configuring H.264 Packetization Mode
  - Monitoring Reference files or Intra Frames
- Verifying Additional Configurations for Video Recording

### TDoS Attack Mitigation

- Finding Feature Information
CHAPTER 13
Cisco Unified Communications Gateway Services—Extended Media Forking 125
Feature Information for Cisco Unified Communications Gateway Services—Extended Media Forking 125
Restrictions for Unified Communications Gateway Services—Extended Media Forking 126
Information About Cisco Unified Communications Gateway Services 126
Extended Media Forking (XMF) Provider and XMF Connection 126
XMF Call-Based Media Forking 127
XMF Connection-Based Media Forking 127
Cisco UC Gateway Services Media Forking API with Survivability TCL 128
Media Forking for SRTP Calls 129
Crypto Tag 129
Example of SDP Data sent in an SRTP Call 130
Multiple XMF Applications and Recording Tone 130
Forking Preservation 132
How to Configure UC Gateway Services 133
Configuring Cisco Unified Communication IOS Services on the Device 133
Configuring the XMF Provider 136
Verifying the UC Gateway Services 137
Troubleshooting Tips 139
Configuration Examples for UC Gateway Services 140
Example: Configuring Cisco Unified Communication IOS Services 140
Example: Configuring the XMF Provider 140
Example: Configuring UC Gateway Services 140

CHAPTER 14
Dynamic Payload Type Interworking for DTMF and Codec Packets for SIP-to-SIP Calls 141
Feature Information for Dynamic Payload Type Interworking for DTMF and Codec Packets for SIP-to-SIP Calls 141
Restrictions for Dynamic Payload Type Interworking for DTMF and Codec Packets for SIP-to-SIP Calls 142
Symmetric and Asymmetric Calls 143
High Availability Checkpointing Support for Asymmetric Payload 144
How to Configure Dynamic Payload Type Passthrough for DTMF and Codec Packets for SIP-to-SIP Calls 144
Configuring Dynamic Payload Type Passthrough at the Global Level 144
Configuring Dynamic Payload Type Passthrough for a Dial Peer 145
Verifying Dynamic Payload Interworking for DTMF and Codec Packets Support 147
Troubleshooting Tips 147
Configuration Examples for Asymmetric Payload Interworking 148
Example: Asymmetric Payload Interworking—Passthrough Configuration 148
Example: Asymmetric Payload Interworking—Interworking Configuration 149

CHAPTER 15

Acoustic Shock Protection 151
Finding Feature Information 151
Restrictions for ASP 151
Information About ASP 152
Acoustic Shock Protection 152
How to Configure ASP 153
Creating the Media Profile for ASP 153
Creating the Media Profile to Enable ASP 154
Configuring the Media Class at a Dial Peer Level for ASP 155
Configuring the Media Class Globally for ASP 156
Verifying ASP 157
Troubleshooting Tips 158
Configuration Examples for the Acoustic Shock Protection Feature 158
Feature Information for Acoustic Shock Protection 159

CHAPTER 16

Noise Reduction 161
Finding Feature Information 161
Prerequisites for Noise Reduction 161
Restrictions for NR 162
Information About NR  162
Noise Reduction  162
How to Configure NR  163
    Creating the Media Profile for NR  163
    Creating the Media Class to Enable NR  164
    Configuring the Media Class at a Dial Peer Level for NR  165
    Configuring the Media Class Globally for NR  166
    Verifying NR  167
    Troubleshooting Tips  168
Configuration Examples for the NR feature  168
Feature Information for Noise Reduction  169

CHAPTER 17

iLBC Support for SIP and H.323  171
    Finding Feature Information  171
    Prerequisites for iLBC Support for SIP and H.323  171
    Restrictions for iLBC Support for SIP and H.323  172
    Information About iLBC Support for SIP and H.323  172
    How to Configure an iLBC Codec  172
        Configuring an iLBC Codec on a Dial Peer  172
        Configuring an iLBC Codec in the Voice Class  174
        Verifying iLBC Support for SIP and H.323  175
    Feature Information for iLBC Support for SIP and H.323  176

CHAPTER 18

Configuring RTP Media Loopback for SIP Calls  179
    Finding Feature Information  179
    Prerequisites  179
    Restrictions  180
    Information About RTP Media Loopback for SIP Calls  180
    How to Configure RTP Media Loopback for SIP Calls  180
    Configuration Examples for RTP Media Loopback  182
        Example: Configuring Video Loopback with Cisco Telepresence System  182
        Example: Configuring Video Loopback with Cisco Unified Video Advantage  182
    Feature Information for RTP Media Loopback for SIP Calls  183

CHAPTER 19

SIP Ability to Send a SIP Registration Message on a Border Element  185
Finding Feature Information 185
Prerequisites for SIP Ability to Send a SIP Registration Message on a Border Element 185
Configuring SIP Ability to Send a SIP Registration Message on a Border Element 186
Feature Information for Sending a SIP Registration Message from a Cisco Unified Border Element 187

CHAPTER 20  Session Refresh with Reinvites 189
Finding Feature Information 189
Prerequisites for Session Refresh with Reinvites 189
Information about Session Refresh with Reinvites 190
How to Configure Session Refresh with Reinvites 190
Configuring Session refresh with Reinvites 190
Feature Information for Session Refresh with Reinvites 192

CHAPTER 21  SIP Stack Portability 193
Finding Feature Information 193
Prerequisites for SIP Stack Portability 193
Information About SIP Stack Portability 194
SIP Call-Transfer Basics 194
  Basic Terminology of SIP Call Transfer 194
  Types of SIP Call Transfer Using the Refer Message Request 197
Feature Information for SIP Stack Portability 205

CHAPTER 22  Interworking of Secure RTP calls for SIP and H.323 207
Finding Feature Information 207
Prerequisites for Interworking of Secure RTP calls for SIP and H.323 208
Restrictions for Interworking of Secure RTP calls for SIP and H.323 208
Feature Information for Configuring Interworking of Secure RTP Calls for SIP and H.323 209

CHAPTER 23  Cisco UBE Support for SRTP-RTP Internetworking 211
Prerequisites for CUBE Support for SRTP-RTP Internetworking 211
Restrictions for CUBE Support for SRTP-RTP Internetworking 212
Information About CUBE for SRTP-RTP Internetworking 212
  CUBE Support for SRTP-RTP Internetworking 212
  TLS on the Cisco Unified Border Element 214
Supplementary Services Support on the Cisco UBE for RTP-SRTP Calls 214
How to Configure Cisco UBE Support for SRTP-RTP Internetworking 215
Configuring Cisco UBE Support for SRTP-RTP Internetworking 215
   Configuring the Certificate Authority 215
   Configuring a Trustpoint for the Secure Universal Transcoder 217
   Configuring DSP Farm Services 219
   Associating SCCP to the Secure DSP Farm Profile 220
   Registering the Secure Universal Transcoder to the CUBE 223
   Configuring SRTP-RTP Internetworking Support 226
      Troubleshooting Tips 229
   Enabling SRTP on the Cisco UBE 229
      Enabling SRTP Globally 229
      Enabling SRTP on a Dial Peer 230
      Troubleshooting Tips 232
   Verifying SRTP-RTP Supplementary Services Support on the Cisco UBE 232
   Configuration Examples for CUBE Support for SRTP-RTP Internetworking 233
      SRTP-RTP Internetworking Example 233
   Feature Information for CUBE Support for SRTP-RTP Internetworking 235

CHAPTER 24
Support for SRTP Termination 237
   Finding Feature Information 237
   Information About Support for SRTP Termination 237
      For End Devices Supporting AES_CM_128_HMAC_SHA1_80 Crypto Suite 238
      For End Devices Supporting AES_CM_128_HMAC_SHA1_32 Crypto Suite 239
   How to Configure Support for SRTP Termination 240
      Configuring Crypto Authentication 240
         Configuring Crypto Authentication (Global Level) 240
         Configuring Crypto Authentication (Dial Peer Level) 241
   Verifying Support for SRTP Termination 242
   Configuration Examples for Support for SRTP Termination 243
      Example: Configuring Crypto Authentication 243
         Example: Configuring Crypto Authentication (Global Level) 243
         Example: Configuring Crypto Authentication (Dial Peer Level) 243
   Additional References for Support for SRTP Termination 244
   Feature Information for Support for SRTP Termination 244
CHAPTER 28  VoIP for IPv6 273
   Prerequisites 273
   Configuring VoIP for IPv6 273
   Feature Information for VoIP for IPv6 274

CHAPTER 29  Mid-call Signaling Consumption 275
   Feature Information for Mid-call Signaling 275
   Prerequisites 276
   Mid-call Signaling Passthrough - Media Change 277
      Restrictions for Mid-call Signaling Passthrough - Media Change 277
      Behavior of Mid-call Re-INVITE Consumption 277
      Configuring Passthrough of Mid-call Signalling 278
      Example Configuring Passthrough SIP Messages at Dial Peer Level 280
      Example Configuring Passthrough SIP Messages at the Global Level 280
   Mid-call Signaling Block 280
      Restrictions for Mid-Call Signaling Block 280
      Blocking Mid-Call Signaling 281
      Example Blocking SIP Messages at Dial Peer Level 282
      Example: Blocking SIP Messages at the Global Level 282
   Mid Call Codec Preservation 282
      Configuring Mid Call Codec Preservation 283
      Example: Configuring Mid Call Codec Preservation at the Dial Peer Level 284
      Example: Configuring Mid Call Codec Preservation at the Global Level 284

CHAPTER 30  Support for Software Media Termination Point 285
   Finding Feature Information 285
   Information About Support for Software Media Termination Point 285
   How to Configure Support for Software Media Termination Point 286
   Prerequisites 286
   Restrictions 286
   Configuring Support for Software Media Termination Point 286
      Examples 289
      Troubleshooting Tips 290
   Feature Information for Support for Software Media Termination Point 291
CHAPTER 35

Fax Detection for SIP Call and Transfer 325
  Finding Feature Information 325
  Restrictions for Fax Detection for SIP Call and Transfer 325
  Information About Fax Detection for SIP Call and Transfer 326
    Mode 1—Local Redirect 326
    Mode 2—Refer Redirect 327
  How to Configure Fax Detection for SIP Calls 328
    Enabling CNG Fax Detection 328
    Verifying Fax Detection for SIP Calls 329
    Troubleshooting Fax Detection for SIP Calls 330
  Configuration Examples for Fax Detection for SIP Calls 330
    Example: Configuring Local Redirect 330
    Example: Configuring Refer Redirect 331
  Feature Information for Fax Detection for SIP Call and Transfer 331
Cisco Unified Border Element
Protocol-Independent Features and Setup

This Cisco Unified Border Element is a special Cisco IOS software image 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
For the latest feature information and caveats, see 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 "Cisco Unified Border Element Features Roadmap".

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

Cisco Unified Border Element Protocol-Independent Features and Setup

This chapter contains the following configuration topics:
Cisco UBE Prerequisites and Restrictions

- Prerequisites for Cisco Unified Border Element
- Restrictions for Cisco Unified Border Element

Dial Plan Management

- Dial Peer Configuration on Voice Gateway Routers
  
  - Translation Rules
  
  - ENUM Support
    - Configuring Tool Command Language (Tcl)
      
  - Cisco Service Advertisement Framework (SAF)
    

Configuring Call Admission Control (CAC)

- VoIP Call Admissions Control
  
  - VoIP Call Admission Control Using RSVP
    

RSVP

- Configuring RSVP Agent
  - Interworking Between RSVP Capable and RSVP Incapable Networks

Dual-Tone Multifrequency (DTMF) Support and Interworking

- SIP--INFO Method for DTMF Tone Generation
- DTMF Events through SIP Signaling
- Configuring SIP DTMF Features
  
  - H.323 RFC2833 - SIP NOTIFY
    
Codec Negotiation

- Support for Negotiation of an Audio Codec from a List of Codecs on Each Leg of a SIP-to-SIP Call on the Cisco Unified Border Element

Payload Type Interoperability

- Dynamic payload type interworking for DTMF and codec packets for SIP-to-SIP calls

Transcoding

- iLBC Support for SIP and H.323
- Universal Transcoding

Fax/modem Support

- Modem Passthrough
- T.38 Fax Relay

- Cisco Fax Relay


SIP Video

- SIP Video Calls with Flow Around Media
- RTP Media Loopback for SIP Calls
- Configuring RTP Media Loopback for SIP Calls

Telepresence

- SIP Video Support for Telepresence Calls

Security Features

- Toll Fraud Prevention

- Access lists (ACLs)

- CAC (call spike)

- SIP—Ability to Send a SIP Registration Message on a Border Element
• SIP Parameter Modification
• SIP—SIP Stack Portability
• Session Refresh with Reinvites
• CDR


• Transport Layer Security (TLS)
• Interworking of Secure RTP calls for SIP and H.323
• SIP SRTP Fallback to Nonsecure RTP
• Cisco Unified Communications Trusted Firewall

**IPv4 and IPv6 Interworking**

• VoIP for IPv6
  • IPv4 to IPv6 Calls (SIP and SIP)
  • IPv6 to IPv6 Calls (SIP and SIP)
  • Support for Dual Stack ANAT

**RSVP Interworking**

• Support for Interworking Between RSVP Capable and RSVP Incapable Networks

**Collocated Services**

• Media Termination Point (MTP)
• Cisco Unified SIP Survivable Remote Site Telephony (SRST)
• Cisco IOS Tcl IVR and VoiceXML Application Guide
• Cisco VoiceXML Programmer’s Guide
• Cisco Unified Communications Trusted Firewall
• Cisco Unified Border Element with Gatekeeper


**Toll Fraud Prevention**

When a Cisco router platform is installed with a voice-capable Cisco IOS software image, appropriate features must be enabled on the platform to prevent potential toll fraud exploitation by unauthorized users. Deploy these features on all Cisco router Unified Communications applications that process voice calls, such as Cisco Unified Communications Manager Express (CME), Cisco Survivable Remote Site Telephony (SRST), Cisco Unified Border Element (UBE), Cisco IOS-based router and standalone analog and digital PBX and
public-switched telephone network (PSTN) gateways, and Cisco contact-center VoiceXML gateways. These features include, but are not limited to, the following:

- Disable secondary dial tone on voice ports--By default, secondary dial tone is presented on voice ports on Cisco router gateways. Use private line automatic ringdown (PLAR) for foreign exchange office (FXO) ports and direct-inward-dial (DID) for T1/E1 ports to prevent secondary dial tone from being presented to inbound callers.

- Cisco router access control lists (ACLs)--Define ACLs to allow only explicitly valid sources of calls to the router or gateway, and therefore to prevent unauthorized Session Initiation Protocol (SIP) or H.323 calls from unknown parties to be processed and connected by the router or gateway.

- Close unused SIP and H.323 ports--If either the SIP or H.323 protocol is not used in your deployment, close the associated protocol ports. If a Cisco voice gateway has dial peers configured to route calls outbound to the PSTN using either time division multiplex (TDM) trunks or IP, close the unused H.323 or SIP ports so that calls from unauthorized endpoints cannot connect calls. If the protocols are used and the ports must remain open, use ACLs to limit access to legitimate sources.

- Change SIP port 5060--If SIP is actively used, consider changing the port to something other than well-known port 5060.

- SIP registration--If SIP registration is available on SIP trunks, turn on this feature because it provides an extra level of authentication and validation that only legitimate sources can connect calls. If it is not available, ensure that the appropriate ACLs are in place.

- SIP Digest Authentication--If the SIP Digest Authentication feature is available for either registrations or invites, turn this feature on because it provides an extra level of authentication and validation that only legitimate sources can connect calls.

- Explicit incoming and outgoing dial peers--Use explicit dial peers to control the types and parameters of calls allowed by the router, especially in IP-to-IP connections used on CME, SRST, and Cisco UBE. Incoming dial peers offer additional control on the sources of calls, and outgoing dial peers on the destinations. Incoming dial peers are always used for calls. If a dial peer is not explicitly defined, the implicit dial peer 0 is used to allow all calls.

- Explicit destination patterns--Use dial peers with more granularity than T for destination patterns to block disallowed off-net call destinations. Use class of restriction (COR) on dial peers with specific destination patterns to allow even more granular control of calls to different destinations on the PSTN.

- Translation rules--Use translation rules to manipulate dialed digits before calls connect to the PSTN to provide better control over who may dial PSTN destinations. Legitimate users dial an access code and an augmented number for PSTN for certain PSTN (for example, international) locations.

- Tcl and VoiceXML scripts--Attach a Tcl/VoiceXML script to dial peers to do database lookups or additional off-router authorization checks to allow or deny call flows based on origination or destination numbers. Tcl/VoiceXML scripts can also be used to add a prefix to inbound DID calls. If the prefix plus DID matches internal extensions, then the call is completed. Otherwise, a prompt can be played to the caller that an invalid number has been dialed.

- Host name validation--Use the "permit hostname" feature to validate initial SIP Invites that contain a fully qualified domain name (FQDN) host name in the Request Uniform Resource Identifier (Request URI) against a configured list of legitimate source hostnames.

- Dynamic Domain Name Service (DNS)--If you are using DNS as the "session target" on dial peers, the actual IP address destination of call connections can vary from one call to the next. Use voice source groups and ACLs to restrict the valid address ranges expected in DNS responses (which are used subsequently for call setup destinations).
For more configuration guidance, see the "Cisco IOS Unified Communications Toll Fraud Prevention" paper.
Interworking Between RSVP Capable and RSVP Incapable Networks

The Interworking Between RSVP Capable and RSVP Incapable Networks feature provides precondition-based Resource Reservation Protocol (RSVP) support for basic audio call and supplementary services on Cisco Unified Border Element (UBE). This feature improves the interoperability between RSVP and non-RSVP networks. RSVP functionality added to Cisco UBE helps you to reserve the required bandwidth before making a call.

This feature extends RSVP support to delayed-offer to delayed-offer and delayed-offer to early-offer calls, along with the early-offer to early-offer calls.

• Finding Feature Information, page 7
• Prerequisites for Interworking Between RSVP Capable and RSVP Incapable Networks, page 8
• Restrictions for Interworking Between RSVP Capable and RSVP Incapable Networks, page 8
• How to Configure Interworking Between RSVP Capable and RSVP Incapable Networks, page 8
• Troubleshooting for Interworking Between RSVP Capable and RSVP Incapable Networks Feature, page 17
• Verifying Interworking Between RSVP Capable and RSVP Incapable Networks, page 18
• Feature Information for Interworking Between RSVP Capable and RSVP Incapable Networks, page 19

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.
Prerequisites for Interworking Between RSVP Capable and RSVP Incapable Networks

- RSVP policies allow you to configure separate bandwidth pools with varying limits so that any one application, such as video, can consume all the RSVP bandwidth on a specified interface at the expense of other applications, such as voice, which would be dropped.
- To limit bandwidth per application, you must configure a bandwidth limit before configuring Support for the Interworking Between RSVP Capable and RSVP Incapable Networks feature. See the Configuring RSVP on an Interface, on page 8.

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.

Restrictions for Interworking Between RSVP Capable and RSVP Incapable Networks

The Support for Interworking Between RSVP Capable and RSVP Incapable Networks feature has the following restrictions:

- Segmented RSVP is not supported.
- Interoperability between Cisco UBE and Cisco Unified Communications Manager is not available.
- RSVP-enabled video calls are not supported.

How to Configure Interworking Between RSVP Capable and RSVP Incapable Networks

Configuring RSVP on an Interface

You must allocate some bandwidth for the interface before enabling RSVP. Perform this task to configure RSVP on an interface.
SUMMARY STEPS

1. enable
2. configure terminal
3. interface type slot / port
4. ip rsvp bandwidth [reservable-bw [max-reservable-bw] [sub-pool reservable-bw]]
5. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode. • Enter your password if prompted.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> interface type slot / port</td>
<td>Configures an interface type and enters interface configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device(config)# interface FastEthernet 0/1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> ip rsvp bandwidth [reservable-bw [max-reservable-bw] [sub-pool reservable-bw]]</td>
<td>Enables RSVP for IP on an interface.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device(config-if)# ip rsvp bandwidth 10000 100000</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> end</td>
<td>(Optional) Exits interface configuration mode and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device(config-if)# end</td>
<td></td>
</tr>
</tbody>
</table>

Configuring Optional RSVP on the Dial Peer

Perform this task to configure optional RSVP at the dial peer level. This configuration allows you to have uninterrupted call even if there is a failure in bandwidth reservation.
### SUMMARY STEPS

1. enable  
2. configure terminal  
3. dial-peer voice *tag* voip  
4. no acc-qos {controlled-load | guaranteed-delay} [audio | video]  
5. req-qos {controlled-load | guaranteed-delay} [audio | video] [bandwidth [default bandwidth-value] [max bandwidth-value]]  
6. end

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **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 77 voip | Enters dial peer voice configuration mode.  
| **Step 4**  
**no acc-qos {controlled-load | guaranteed-delay} [audio | video]**  
*Example:*  
Device(config-dial-peer)# no acc-qos controlled-load | Removes any value configured for the acc-qos command.  
*Keywords are as follows:*  
• **controlled-load**--Indicates that RSVP guarantees a single level of preferential service, presumed to correlate to a delay boundary. The controlled load service uses admission (or capacity) control to ensure that preferential service is received even when the bandwidth is overloaded.  
• **guaranteed-delay**--Indicates that RSVP reserves bandwidth and guarantees a minimum bit rate and preferential queueing if the bandwidth reserved is not exceeded.  
| **Step 5**  
**req-qos {controlled-load | guaranteed-delay} [audio | video] [bandwidth [default bandwidth-value] [max bandwidth-value]]** | Configures the desired quality of service (QoS) to be used.  
*Calls continue even if there is a failure in bandwidth reservation.*  

Configure Mandatory RSVP on the Dial Peer

Perform this task to configure Mandatory RSVP on the dial peer. This configuration ensures that the call does not connect if sufficient bandwidth is not allocated.

SUMMARY STEPS

1. enable
2. configure terminal
3. dial-peer voice tag voip
4. acc-qos {best-effort | controlled-load | guaranteed-delay} [audio | video]
5. req-qos {best-effort [audio | video] | {controlled-load | guaranteed-delay} [audio | video] [bandwidth [default bandwidth-value] [max bandwidth-value]]}
6. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example: Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>Command or Action</td>
</tr>
<tr>
<td>--------</td>
<td>------------------</td>
</tr>
<tr>
<td></td>
<td><strong>dial-peer voice tag voip</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> Device(config)# dial-peer 77 voip</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 4</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>**acc-qos {best-effort</td>
<td>controlled-load</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> Device(config-dial-peer)# acc-qos best-effort</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Keywords are as follows:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• <strong>best-effort</strong>--Indicates that Resource Reservation Protocol (RSVP) makes no bandwidth reservation. This is the default.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• <strong>controlled-load</strong>--Indicates that RSVP guarantees a single level of preferential service, presumed to correlate to a delay boundary. The controlled load service uses admission (or capacity) control to ensure that preferential service is received even when the bandwidth is overloaded.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• <strong>guaranteed-delay</strong>--Indicates that RSVP reserves bandwidth and guarantees a minimum bit rate and preferential queueing if the bandwidth reserved is not exceeded.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 5</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>**req-qos {best-effort [audio</td>
<td>video]}</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> Device(config-dial-peer)# req-qos controlled-load</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Calls continue even if there is a drop in the bandwidth reservation.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 6</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>end</strong></td>
<td>(Optional) Exits dial peer voice configuration mode and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> Device(config-dial-peer)# end</td>
<td></td>
</tr>
</tbody>
</table>

---

**Configuring Midcall RSVP Failure Policies**

Perform this task to enable call handling policies for a midcall RSVP failure.
**SUMMARY STEPS**

1. enable
2. configure terminal
3. dial-peer voice `tag voip`
4. `voice-class sip rsvp-fail-policy {video | voice} post-alert {optional keep-alive | mandatory {keep-alive | disconnect retry retry-attempts}} interval seconds`
5. end

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example: Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> dial-peer voice <code>tag voip</code></td>
<td>Enters dial peer voice configuration mode.</td>
</tr>
<tr>
<td>Example: Device(config)# dial-peer voice 66 voip</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> voice-class sip rsvp-fail-policy `{video</td>
<td>voice}<code>post-alert</code>{optional keep-alive</td>
</tr>
<tr>
<td>Example: Device(config-dial-peer)# voice-class sip rsvp-fail-policy voice post-alert mandatory keep-alive interval 50</td>
<td>Keywords are as follows:</td>
</tr>
<tr>
<td></td>
<td>• optional keep-alive--The keepalive messages are sent when RSVP fails only if RSVP negotiation is optional.</td>
</tr>
<tr>
<td></td>
<td>• mandatory keep-alive--The keepalive messages are sent when RSVP fails only if RSVP negotiation is mandatory.</td>
</tr>
<tr>
<td><strong>Note</strong> Keepalive messages are sent at 30-second intervals when a postalert call fails to negotiate RSVP regardless of the RSVP negotiation setting (mandatory or optional).</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> end</td>
<td>(Optional) Exits dial peer voice configuration mode and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td>Example: Device(config-dial-peer)# end</td>
<td></td>
</tr>
</tbody>
</table>
Configuring DSCP Values

Perform this task to configure different Differentiated Services Code Point (DSCP) values based on RSVP status.

SUMMARY STEPS

1. enable
2. configure terminal
3. dial-peer voice tag voip
5. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>enable</td>
</tr>
<tr>
<td>Example:</td>
<td>Device&gt; enable</td>
</tr>
<tr>
<td></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>configure terminal</td>
</tr>
<tr>
<td>Example:</td>
<td>Device# configure terminal</td>
</tr>
<tr>
<td></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>dial-peer voice tag voip</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config)# dial-peer voice 66 voip</td>
</tr>
<tr>
<td></td>
<td>Enters dial peer voice configuration mode.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>ip qos dscp {dscp-value</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config-dial-peer)# ip qos dscp af11 media rsvp-pass</td>
</tr>
<tr>
<td></td>
<td>Configures DSCP values based on RSVP status.</td>
</tr>
<tr>
<td></td>
<td>• Keywords are as follows:</td>
</tr>
<tr>
<td></td>
<td>• media rsvp-pass--Specifies that the DSCP value applies to media packets with successful RSVP reservations.</td>
</tr>
<tr>
<td></td>
<td>• media rsvp-fail--Specifies that the DSCP value applies to packets (media or video) with failed RSVP reservations.</td>
</tr>
</tbody>
</table>
### Purpose

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>• The default DSCP value for all media (voice and fax) packets is <strong>ef</strong>.</td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong> You must configure the DSCP values for all cases: <strong>media rsvp-pass</strong> and <strong>media rsvp-fail</strong>.</td>
<td></td>
</tr>
</tbody>
</table>

### Step 5

| end | (Optional) Exits dial peer voice configuration mode and returns to privileged EXEC mode. |

#### Example:

```
Device(config-dial-peer)# end
```

---

### Configuring an Application ID

Perform this task to configure a specific application ID for RSVP establishment.

#### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice tag voip**
4. **ip qos policy-locator {video | voice} [app app-string] [guid guid-string] [sapp subapp-string] [ver version-string]**
5. **end**

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
</tbody>
</table>

#### Example:

```
Device> enable
Device# configure terminal
```
### Configuring Priority

Perform this task to configure priorities for call preemption.

**SUMMARY STEPS**

1. `enable`  
2. `configure terminal`  
3. `dial-peer voice tag voip`  
4. `ip qos defending-priority defending-pri-value`  
5. `ip qos preemption-priority preemption-pri-value`  
6. `end`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><code>enable</code></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Enter your password if prompted.</td>
</tr>
<tr>
<td><code>Device&gt; enable</code></td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Step 2</th>
<th>configure terminal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Device# configure terminal</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 3</th>
<th>dial-peer voice tag voip</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Device(config)# dial-peer voice 66 voip</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 4</th>
<th>ip qos defending-priority defending-pri-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Device(config-dial-peer)# ip qos defending-priority 66</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 5</th>
<th>ip qos preemption-priority preemption-pri-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Device(config-dial-peer)# ip qos preemption-priority 75</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 6</th>
<th>end</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td>Device(config-dial-peer)# end</td>
</tr>
</tbody>
</table>

### Troubleshooting for Interworking Between RSVP Capable and RSVP Incapable Networks Feature

Use the following commands to debug any errors that you may encounter when you configure the Support for Interworking Between RSVP Capable and RSVP Incapable Networks feature.

- debug call rsvp-sync events
- debug call rsvp-sync func-trace
- debug ccsip all
- debug ccsip messages
- debug ip rsvp messages
- debug sccp all
Verifying Interworking Between RSVP Capable and RSVP Incapable Networks

This task explains how to display information to verify the configuration for the Support for Interworking Between RSVP Capable and RSVP Incapable Networks feature. These commands need not be entered in any specific order.

SUMMARY STEPS

1. enable
2. show sip-ua calls
3. show ip rsvp installed
4. show ip rsvp reservation
5. show ip rsvp interface detail [interface-type number]
6. show sccp connections details
7. show sccp connections rsvp
8. show sccp connections internal
9. show sccp [all | connections | statistics]

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device&gt; enable</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong> show sip-ua calls</td>
<td>(Optional) Displays active user agent client (UAC) and user agent server (UAS) information on SIP calls.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device# show sip-ua calls</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> show ip rsvp installed</td>
<td>(Optional) Displays RSVP-related installed filters and corresponding bandwidth information.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device# show ip rsvp installed</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> show ip rsvp reservation</td>
<td>(Optional) Displays RSVP-related receiver information currently in the database.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device# show ip rsvp reservation</td>
<td></td>
</tr>
</tbody>
</table>
### Feature Information for Interworking Between RSVP Capable and RSVP Incapable Networks

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 . An account on Cisco.com is not required. Feature History Table entry for the Cisco Unified Border Element.

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 5</strong> show ip rsvp interface detail [interface-type number]</td>
<td>(Optional) Displays the interface configuration for hello.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device# show ip rsvp interface detail GigabitEthernet 0/0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> show sccp connections details</td>
<td>(Optional) Displays SCCP connection details, such as call-leg details.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device# show sccp connections details</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> show sccp connections rsvp</td>
<td>(Optional) Displays information about active SCCP connections that are using RSVP.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device# show sccp connections rsvp</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> show sccp connections internal</td>
<td>(Optional) Displays the internal SCCP details, such as time-stamp values.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device# show sccp connections internal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong> show sccp [all</td>
<td>connections</td>
</tr>
<tr>
<td><strong>Example:</strong> Device# show sccp statistics</td>
<td></td>
</tr>
</tbody>
</table>
### Table 1: Feature Information for Interworking Between RSVP Capable and RSVP Incapable Networks

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interworking Between RSVP Capable and RSVP Incapable Networks</td>
<td>15.0(1)XA 15.1(1)T</td>
<td>The Interworking Between RSVP Capable and RSVP Incapable Networks feature provides precondition-based RSVP support for basic audio call and supplementary services on the Cisco UBE. The following commands were introduced or modified: acc-qos, ip qos defending-priority, ip qos dscp, ip qos policy-locator, ip qos preemption-priority, req-qos, voice-class sip rsvp-fail-policy,</td>
</tr>
<tr>
<td>Interworking Between RSVP Capable and RSVP Incapable Networks</td>
<td>Cisco IOS XE Release 3.1S</td>
<td>The Interworking Between RSVP Capable and RSVP Incapable Networks feature provides precondition-based RSVP support for basic audio call and supplementary services on the Cisco UBE. The following commands were introduced or modified: acc-qos, ip qos defending-priority, ip qos dscp, ip qos policy-locator, ip qos preemption-priority, req-qos, voice-class sip rsvp-fail-policy,</td>
</tr>
</tbody>
</table>
CHAPTER 3

SIP INFO Method for DTMF Tone Generation

The SIP: INFO Method for DTMF Tone Generation feature uses the Session Initiation Protocol (SIP) INFO method to generate dual tone multifrequency (DTMF) tones on the telephony call leg. SIP info methods, or request message types, request a specific action be taken by another user agent (UA) or proxy server. The SIP INFO message is sent along the signaling path of the call. Upon receipt of a SIP INFO message with DTMF relay content, the gateway generates the specified DTMF tone on the telephony end of the call.

- Finding Feature Information, page 21
- Prerequisites for SIP INFO Method for DTMF Tone Generation, page 21
- Restrictions for SIP INFO Methods for DTMF Tone Generation, page 22
- Information About SIP INFO Method for DTMF Tone Generation, page 22
- How to Review SIP INFO Messages, page 22
- Configuring for SIP INFO Method for DTMF Tone Generation, page 23
- Troubleshooting Tips, page 23
- Feature Information for SIP INFO Method for DTMF Tone Generation, page 24

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for SIP INFO Method for DTMF Tone Generation

You cannot configure, enable, or disable this feature. No configuration tasks are required to configure the SIP - INFO Method for DTMF Tone Generation feature. The feature is enabled by default.
Cisco Unified Border Element

- Cisco IOS Release 12.2(11)T or a later release must be installed and running on your Cisco Unified Border Element.

Cisco Unified Border Element (Enterprise)

- Cisco IOS XE Release 2.5 or a later release must be installed and running on your Cisco ASR 1000 Series Router.

Restrictions for SIP INFO Methods for DTMF Tone Generation

The SIP: INFO Method for DTMF Tone Generation feature includes the following signal duration parameters:

- Minimum signal duration is 100 milliseconds (ms). If a request is received with a duration less than 100 ms, the minimum duration of 100 ms is used by default.
- Maximum signal duration is 5000 ms. If a request is received with a duration longer than 5000 ms, the maximum duration of 5000 ms is used by default.
- If no duration parameter is included in a request, the gateway defaults to a signal duration of 250 ms.

Information About SIP INFO Method for DTMF Tone Generation

The SIP: INFO Method for DTMF Tone Generation feature is always enabled, and is invoked when a SIP INFO message is received with DTMF relay content. This feature is related to the DTMF Events Through SIP Signaling feature, which allows an application to be notified about DTMF events using SIP NOTIFY messages. Together, the two features provide a mechanism to both send and receive DTMF digits along the signaling path. For more information on sending DTMF event notification using SIP NOTIFY messages, refer to the DTMF Events Through SIP Signaling feature.

How to Review SIP INFO Messages

The SIP INFO method is used by a UA to send call signaling information to another UA with which it has an established media session. The following example shows a SIP INFO message with DTMF content:

```
INFO sip:2143302100@172.17.2.33 SIP/2.0
Via: SIP/2.0/UDP 172.80.2.100:5060
From: <sip:9724401003@172.80.2.100>;tag=43
To: <sip:2143302100@172.17.2.33>;tag=9753.0207
Call-ID: 984072_15401962@172.80.2.100
CSeq: 25634 INFO
Supported: 100rel
Supported: timer
Content-Length: 26
Content-Type: application/dtmf-relay
Signal= 1
Duration= 160
```

This sample message shows a SIP INFO message received by the gateway with specifics about the DTMF tone to be generated. The combination of the "From", "To", and "Call-ID" headers identifies the call leg. The
signal and duration headers specify the digit, in this case 1, and duration, 160 milliseconds in the example, for DTMF tone play.

**Configuring for SIP INFO Method for DTMF Tone Generation**

You cannot configure, enable, or disable this feature. No configuration tasks are required to configure the SIP-INFO Method for DTMF Tone Generation feature. The feature is enabled by default.

**Troubleshooting Tips**

You can display SIP statistics, including SIP INFO method statistics, by using the `show sip-ua statistics` and `show sip-ua status` commands in privileged EXEC mode. See the following fields for SIP INFO method statistics:

- **OkInfo 0/0**, under SIP Response Statistics, Success, displays the number of successful responses to an INFO request.
- **Info 0/0**, under SIP Total Traffic Statistics, displays the number of INFO messages received and sent by the gateway.

The following is sample output from the `show sip-ua statistics` command:

```
Device# show sip-ua statistics
SIP Response Statistics (Inbound/Outbound)
  Informational: Trying 1/1, Ringing 0/0,
  Forwarded 0/0, Queued 0/0,
  SessionProgress 0/1
  Success: OkInvite 0/1, OkBye 1/0,
  OkCancel 0/0, OkOptions 0/0,
  OkPrack 0/0, OkPreconditionMet 0/0
  OkSubscribe 0/0, OkNotify 0/0,
  OkInfo 0/0, 202Accepted 0/0
  Redirection (Inbound only):
    MultipleChoice 0, MovedPermanently 0,
    MovedTemporarily 0, SeeOther 0,
    UseProxy 0, AlternateService 0
  Client Error:
    BadRequest 0/0, Unauthorized 0/0,
    PaymentRequired 0/0, Forbidden 0/0,
    NotFound 0/0, MethodNotAllowed 0/0,
    NotAcceptable 0/0, ProxyAuthRequired 0/0,
    ReqTimeout 0/0, Conflict 0/0, Gone 0/0,
    LengthRequired 0/0, ReqEntityTooLarge 0/0,
    ReqURITooLarge 0/0, UnsupportedMediaType 0/0,
    BadExtension 0/0, TempNotAvailable 0/0,
    CallLegNonExisting 0/0, LoopDetected 0/0,
    TooManyHops 0/0, AddrIncomplete 0/0,
    Ambiguous 0/0, BusyHere 0/0,
    BadEvent 0/0
  Server Error:
    InternalError 0/0, NotImplemented 0/0,
    BadGateway 0/0, ServiceUnavailable 0/0,
    GatewayTimeout 0/0, BadSipVer 0/0
  Global Failure:
    BusyEverywhere 0/0, Decline 0/0,
    NotExistAnywhere 0/0, NotAcceptable 0/0
SIP Total Traffic Statistics (Inbound/Outbound)
  Invite 0/0, Ack 0/0, Bye 0/0,
  Cancel 0/0, Options 0/0,
```
The following is sample output from the `show sip-ua status` command:

```plaintext
Device# show sip-ua status
SIP User Agent Status
SIP User Agent for UDP : ENABLED
SIP User Agent for TCP : ENABLED
SIP User Agent bind status(signaling): DISABLED
SIP User Agent bind status(media): DISABLED
SIP max-forwards : 6
SIP DNS SRV version: 2 (rfc 2782)
SDP application configuration:
  Version line (v=) required
  Owner line (o=) required
  Session name line (s=) required
  Timespec line (t=) required
  Media supported: audio image
  Network types supported: IN
  Address types supported: IP4
  Transport types supported: RTP/AVP udptl
```

### Feature Information for SIP INFO Method for DTMF Tone Generation

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 . An account on Cisco.com is not required.

**Table 2: Feature Information for SIP: INFO Method for DTMF Tone Generation**

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIP: INFO Method for DTMF Tone Generation</td>
<td>12.2(11)T 12.3(2)T 12.2(8)YN</td>
<td>The SIP: INFO Method for DTMF Tone Generation feature uses the Session Initiation Protocol (SIP) INFO method to generate dual-tone multifrequency (DTMF) tones on the telephony call leg. SIP methods, or request message types, request a specific action be taken by another user agent (UA) or proxy server. The SIP INFO message is sent along the signaling path of the call. The following command was introduced: <code>show sip-ua</code>.</td>
</tr>
<tr>
<td></td>
<td>12.2(11)YV 12.2(11)T 12.2(15)T</td>
<td></td>
</tr>
</tbody>
</table>

Cisco Unified Border Element Protocol-Independent Features and Setup Configuration Guide, Cisco IOS Release 15M&T

24
<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIP: INFO Method for DTMF Tone Generation</td>
<td>Cisco IOS XE Release 2.5S</td>
<td>The SIP: INFO Method for DTMF Tone Generation feature uses the Session Initiation Protocol (SIP) INFO method to generate dual-tone multifrequency (DTMF) tones on the telephony call leg. SIP methods, or request message types, request a specific action be taken by another user agent (UA) or proxy server. The SIP INFO message is sent along the signaling path of the call. The following command was introduced: <code>show sip-ua</code>.</td>
</tr>
</tbody>
</table>
WebEx Telepresence Media Support Over Single SIP Session

The WebEx Telepresence Media Support over Single SIP Session feature provides support for end-to-end negotiation of up to 6 m-lines or media lines over a single Session Initiation Protocol (SIP) session. The media types can be audio, video, or application.

- Finding Feature Information, page 27
- Restrictions for WebEx Telepresence Media Support Over Single SIP Session, page 27
- Information About WebEx Telepresence Media Support Over Single SIP Session, page 28
- Monitoring WebEx Telepresence Media Support Over Single SIP Session, page 28
- Feature Information for WebEx Telepresence Media Support Over Single SIP Session, page 31

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for WebEx Telepresence Media Support Over Single SIP Session

- High availability is not supported with multiple m-lines.
- Only single dynamic payload type in the m-line for H.224 protocol is supported.
- Payload type interworking for Aggregation Service Routers (ASR) is not supported, so dynamic payload type is negotiated end-to-end.
Information About WebEx Telepresence Media Support Over Single SIP Session

The WebEx Telepresence Media Support over Single SIP Session feature provides the following support:

- End-to-end negotiation of multiple m-lines.
- Negotiation of Binary Floor Control Protocol (BFCP), IX, and H.224 protocol m-lines (m=application) and creation of Real-time Transport Protocol (RTP) or UDP streams for the same.
- Early-Offter (EO-EO) and Delayed-Offter (DO-DO) calls' support by the Cisco Unified Border Element (Cisco UBE) with multiple m-lines.
- End-to-end negotiation of multiple m-lines of same media type for video and application (but not audio).
- Mid-call escalation and de-escalation for multiple application and video m-lines.
- Secure RTP (SRTP) passthrough for all RTP streams (audio, video, and application).
- SRTP-RTP interworking for video (ASR only).
- Multiple dynamic payload types in the same m-line for the H.264 codec.

You can use the `show voip rtp connections` and `show call active video compact` commands to see the details about additional video and application streams.

Monitoring WebEx Telepresence Media Support Over Single SIP Session

Perform this task to see the details about additional video and application streams. The `show` commands can be entered in any order.

**SUMMARY STEPS**

1. `enable`
2. `show call active video compact`
3. `show voip rtp connections`
4. `show sip-ua calls`

**DETAILED STEPS**

**Step 1** `enable`

Enables privileged EXEC mode.

**Example:**

```
Device> enable
```
Step 2  **show call active video compact**  
Displays a compact version of call information for Skinny Call Control Protocol (SCCP), SIP, and H.323 video calls in progress. The codec type, negotiated codec, and remote media ports are displayed.

**Example:**  
```
Device# show call active video compact
```
```
<callID> A/O FAX T<sec> Codec type Peer Address IP R<ip>:<udp>
Total call-legs:  2
  1 ANS   T5   H264   VOIP-VIDEO  P332211  9.45.38.39:2448
  6 ORG   T5   H264   VOIP-VIDEO  P1111  9.45.38.39:2438
```

Step 3  **show voip rtp connections**  
Displays RTP named event packets. In the following sample output, two RTP connections are displayed for each m-line and a total of 10 RTP connections are displayed for 5 m-lines.

**Example:**  
```
Device# show voip rtp connections
VoIP RTP active connections :
No. CallId dstCallId LocalRTP RmtRTP LocalIP RemoteIP
  1 1  6 16384 54024 192.0.2.123 192.0.2.39
  2 2  7 16386 2448 192.0.2.123 192.0.2.39
  3 3  8 16400 5070 192.0.2.123 192.0.2.39
  4 4  9 16388 2450 192.0.2.123 192.0.2.39
  5 5 10 16402 2452 192.0.2.123 192.0.2.39
  6 6  1 16390 58121 192.0.2.123 192.0.2.39
  7 7  2 16392 2438 192.0.2.123 192.0.2.39
  8 8  3 16394 5070 192.0.2.123 192.0.2.39
  9 9  4 16396 2440 192.0.2.123 192.0.2.39
 10 10  5 16398 2442 192.0.2.123 192.0.2.39
```

Step 4  **show sip-ua calls**  
Displays active user agent client (UAC) and user agent server (UAS) information on Session Initiation Protocol (SIP) calls.

**Example:**  
```
Device# show sip-ua calls
Total SIP call legs: 2, User Agent Client: 1, User Agent Server: 1
SIP UAC CALL INFO
Call 1
SIP Call ID : 72B6C784-753E11E2-FFFFFFFF8008B555-FFFFFFFFE340699E@9.45.47.123
State of the call : STATE_ACTIVE (7)
Substate of the call : SUBSTATE_NONE (0)
Calling Number : 332211
Called Number : 1111
Bit Flags : 0xC04018 0x10000100 0x80
CC Call ID : 6
Source IP Address (Sig ) : 9.45.47.123
Destn SIP Req Addr:Port : [9.45.38.39]:5267
Destn SIP Resp Addr:Port: [9.45.38.39]:5267
Destination Name : 9.45.38.39
Number of Media Streams : 5
Number of Active Streams: 5
RTP Fork Object : 0x0
Media Mode : flow-through
Media Stream 1
State of the stream : STREAM_ACTIVE
Stream Call ID : 6
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: [9.45.47.123]:16390
Media Dest IP Addr:Port : [9.45.38.39]:58121

Media Stream 2
State of the stream : STREAM_ACTIVE
Stream Call ID : 7
Stream Type : video (7)
Stream Media Addr Type : 1
Negotiated Codec : h263 (0 bytes)
Codec Payload Type : 97
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: [9.45.47.123]:16392
Media Dest IP Addr:Port : [9.45.38.39]:2438

Media Stream 3
State of the stream : STREAM_ACTIVE
Stream Call ID : 8
Stream Type : application (8)
Stream Media Addr Type : 1
Negotiated Codec : No Codec (0 bytes)
Codec Payload Type : 255 (None)
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: [9.45.47.123]:16394
Media Dest IP Addr:Port : [9.45.38.39]:5070

Media Stream 4
State of the stream : STREAM_ACTIVE
Stream Call ID : 9
Stream Type : video (7)
Stream Media Addr Type : 1
Negotiated Codec : h263 (0 bytes)
Codec Payload Type : 97
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: [9.45.47.123]:16396
Media Dest IP Addr:Port : [9.45.38.39]:2440

Media Stream 5
State of the stream : STREAM_ACTIVE
Stream Call ID : 10
Stream Type : application (8)
Stream Media Addr Type : 1
Negotiated Codec : H.224 (0 bytes)
Codec Payload Type : 107
Negotiated Dtmf-relay : inband-voice
Dtmf-relay Payload Type : 0
QoS ID : -1
Local QoS Strength : BestEffort
Negotiated QoS Strength : BestEffort
Feature Information for WebEx Telepresence Media Support Over Single SIP Session

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 . An account on Cisco.com is not required.

Table 3: Feature Information for WebEx Telepresence Media Support Over Single SIP Session

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebEx Telepresence Media Support Over Single SIP</td>
<td>15.3(2)T</td>
<td>The WebEx Telepresence Media Support over Single SIP Session feature provides support for end-to-end negotiation of up to 6 m-lines or media lines over a single Session Initiation Protocol (SIP) session. The media types can be audio, video, or application.</td>
</tr>
<tr>
<td>Session</td>
<td>Cisco IOS XE Release 3.9S</td>
<td>The WebEx Telepresence Media Support over Single SIP Session feature provides support for end-to-end negotiation of up to 6 m-lines or media lines over a single Session Initiation Protocol (SIP) session. The media types can be audio, video, or application.</td>
</tr>
</tbody>
</table>
CHAPTER 5

DTMF Events through SIP Signaling

The DTMF Events through SIP Signaling feature provides the following:

- DTMF event notification for SIP messages.
- Capability of receiving hookflash event notification through the SIP NOTIFY method.
- Third-party call control, or other signaling mechanisms, to provide enhanced services, such as calling card and messaging services.
- Communication with the application outside of the media connection.

The DTMF Events through SIP Signaling feature allows telephone event notifications to be sent through SIP NOTIFY messages, using the SIP SUBSCRIBE/NOTIFY method as defined in the Internet Engineering Task Force (IETF) draft, SIP-Specific Event Notification.

The feature also supports sending DTMF notifications based on the IETF draft: Signaled Telephony Events in the Session Initiation Protocol (SIP) (draft-mahy-sip-signaled-digits-01.txt).

- Finding Feature Information, page 33
- Prerequisites for DTMF Events through SIP Signaling, page 34
- Restrictions for DTMF Events through SIP Signaling, page 34
- DTMF Dialing, page 34
- NOTIFY Messages, page 34
- Configuring DTMF Events through SIP Signaling, page 35
- Troubleshooting Tips, page 41
- Feature Information for DTMF Events through SIP Signaling, page 41

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for DTMF Events through SIP Signaling

Cisco Unified Border Element

- Cisco IOS Release 12.2(11)T or a later release must be installed and running on your Cisco Unified Border Element.

Cisco Unified Border Element (Enterprise)

- Cisco IOS XE Release 2.5 or a later release must be installed and running on your Cisco ASR 1000 Series Router.

Restrictions for DTMF Events through SIP Signaling

The DTMF Events through SIP Signaling feature adds support for sending telephone-event notifications via SIP NOTIFY messages from a SIP gateway. The events for which notifications are sent out are DTMF events from the local Plain Old Telephone Service (POTS) interface on the gateway. Notifications are not sent for DTMF events received in the Real-Time Transport Protocol (RTP) stream from the recipient user agent.

DTMF Dialing

DTMF dialing consists of simultaneous voice-band tones generated when a button is pressed on a telephone. The use of DTMF signaling for this feature enables support for advanced telephony services. Currently there are a number of application servers and service creation platforms that do not support media connections. To provide value-added services to the network, these servers and platforms need to be aware of signaling events from a specific participant in the call. Once the server or platform is aware of the DTMF events that are being signaled, it can use third-party call control, or other signaling mechanisms, to provide enhanced services. Examples of the types of services and platforms that are supported by this feature are various voice web browser services, Centrex switches or business service platforms, calling card services, and unified message servers. All of these applications require a method for the user to communicate with the application outside of the media connection. The DTMF Events Through SIP Signaling feature provides this signaling capability. This feature is related to the SIP INFO Method for DTMF Tone Generation feature, which adds support for out-of-band DTMF tone generation using the SIP INFO method. Together the two features provide a mechanism to both send and receive DTMF digits along the signaling path.

NOTIFY Messages

The SIP event notification mechanism uses NOTIFY messages to signal when certain telephony events take place. In order to send DTMF signals through NOTIFY messages, the gateway notifies the subscriber when DTMF digits are signaled by the originator. The notification contains a message body with a SIP response status line.
The following sample message shows a NOTIFY message from the Notifier letting the Subscriber know that the subscription is completed. The combination of the From, To, and Call-ID headers identifies the call leg. The Events header specifies the event type being signaled, and the Content-Type specifies the Internet media type. The Content-Length header indicates the number of octets in the message body.

```
NOTIFY sip:subscriber@example1.com SIP/2.0
Via: SIP/2.0/UDP example2.com:5060
From: Notifier <sip:notifier@example2.com>;tag=5678-EFGH
To: Subscriber <sip:subscriber@example1.com>;tag=1234-ABCD
Call-ID: 12345@example2.com
CSeq: 104 NOTIFY
Contact: Notifier <sip:notifier@example2.com>
Events: telephone-event;rate=1000
Content-Type: audio/telephone-event
Content-Length: 4
```

**Configuring DTMF Events through SIP Signaling**

To configure the DTMF Events through SIP Signaling feature, perform the following steps.

**SUMMARY STEPS**

1. enable
2. configure terminal
3. sip-ua
4. timers notify number
5. retry notify number
6. exit

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters privileged EXEC mode or any other security level set by a system administrator.</td>
</tr>
<tr>
<td>enable</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device&gt; enable</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 2</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device# configure terminal</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 3</strong></td>
<td>Enters SIP user-agent configuration mode.</td>
</tr>
<tr>
<td>sip-ua</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device(config)# sip-ua</td>
<td></td>
</tr>
</tbody>
</table>
Verifying SIP DTMF Support

To verify SIP DTMF support, perform the following steps as appropriate (commands are listed in alphabetical order).

SUMMARY STEPS

1. show running-config
2. show sip-ua retry
3. show sip-ua statistics
4. show sip-ua status
5. show sip-ua timers
6. show voip rtp connections
7. show sip-ua calls

DETAILED STEPS

Step 1  show running-config
Use this command to show dial-peer configurations.

The following sample output shows that the `dtmf-relay sip-notify` command is configured in dial peer 123:

---

**Verifying SIP DTMF Support**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 4</strong></td>
<td>timers notify number</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config-sip-ua)# timers notify 100</td>
</tr>
<tr>
<td></td>
<td>Sets the amount of time that the user agent waits before retransmitting the Notify message. The argument is as follows:</td>
</tr>
<tr>
<td></td>
<td>* number --Time, in milliseconds, to wait before retransmitting. Range: 100 to 1000. Default: 500.</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>retry notify number</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config-sip-ua)# retry notify 6</td>
</tr>
<tr>
<td></td>
<td>Sets the number of times that the Notify message is retransmitted to the user agent that initiated the transfer or Refer request. The argument is as follows:</td>
</tr>
<tr>
<td></td>
<td>* number --Number of retries. Range: 1 to 10. Default: 10.</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>exit</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config-sip-ua)# exit</td>
</tr>
<tr>
<td></td>
<td>Exits the current mode.</td>
</tr>
</tbody>
</table>
Example:

Device# show running-config
#
dial-peer voice 123 voip
destination-pattern {12}...
monitor probe icmp-ping
session protocol sipv2
session target ipv4:10.8.17.42
dtmf-relay sip-notify

The following sample output shows that DTMF relay and NTE are configured on the dial peer.

Example:

Device# show running-config
!
dial-peer voice 1000 pots
destination-pattern 4961234
port 1/0/0
!
dial-peer voice 2000 voip
application session
destination-pattern 4965678
session protocol sipv2
session target ipv4:192.0.2.34
dtmf-relay rtp-nte
! RTP payload type value = 101 (default)
!
dial-peer voice 3000 voip
application session
destination-pattern 2021010101
session protocol sipv2
session target ipv4:192.0.2.34
rtp payload-type nte 110
! RTP payload type value = 110 (user assigned)
!

Step 2 show sip-ua retry
Use this command to display SIP retry statistics.

Example:

Device# show sip-ua retry
SIP UA Retry Values
invite retry count = 6 response retry count = 1
bye retry count = 1 cancel retry count = 1
prack retry count = 10 comet retry count = 10
reliable 1xx count = 6 notify retry count = 10

Step 3 show sip-ua statistics
Use this command to display response, traffic, and retry SIP statistics.

Tip To reset counters for the show sip-ua statistics display, use the clear sip-ua statistics command.

Example:

Device# show sip-ua statistics
SIP Response Statistics (Inbound/Outbound)
Verifying SIP DTMF Support

Following is sample output verifying configuration of the SIP INFO Method for DTMF Tone Generation feature:

Example:

```
Device# show sip-ua statistics
SIP Response Statistics (Inbound/Outbound)
  Informational:
    Trying 1/1, Ringing 0/0,
    Forwarded 0/0, Queued 0/0,
    SessionProgress 0/1
  Success:
    OkInvite 0/1, OkBye 0/0,
    OkCancel 0/0, OkOptions 0/0,
    OkCseq 0/0, OkMaxForwards 0/0,
    OkPreconditionMet 0/0,
    OkSubscibe 0/0, OkNotify 0/0,
    OkInfo 0/0, 200Ok 0/0
  Redirect (Inbound only):
    MultipleChoice 0, MovedPermanently 0,
    MovedTemporarily 0, SeeOther 0,
    UseProxy 0, AlternateService 0
Client Error:
    BadRequest 0/0, Unauthorized 0/0,
    PaymentRequired 0/0, Forbidden 0/0,
    NotFound 0/0, MethodNotAllowed 0/0,
    NotAcceptable 0/0, ProxyAuthReqd 0/0,
    LengthRequired 0/0, Expect 0/0,
    InvalidHeader 0/0, RequestURIEmpty 0/0,
    IllegalParameter 0/0,客户端 Error:
    InternalError 0/0, NotImplemented 0/0,
    ServiceUnavailable 0/0, GatewayTimeout 0/0,
    PreConditionFailure 0/0,
    ServiceProvider 0/0, ServiceUnavailable 0/0
SIP Total Traffic Statistics (Inbound/Outbound) /* Traffic Statistics
  Invite 1/1, Ack 1/1, Prack 0/0
  Cancel 0/1, Options 0/0,
  Prack 0/2, Comet 0/0,
  Notify 0/1, Refer 1/0
```

Cisco Unified Border Element Protocol-Independent Features and Setup Configuration Guide, Cisco IOS Release 15M&T
Step 4

**show sip-ua status**

Use this command to display status for the SIP user agent.

**Example:**

```
Device# show sip-ua status
SIP User Agent Status
SIP User Agent for UDP : ENABLED
SIP User Agent for TCP : ENABLED
SIP User Agent bind status(signaling): DISABLED
SIP User Agent bind status(media): DISABLED
SIP max-forwards : 6
SIP DNS SRV version: 2 (rfc 2782)
SDP application configuration:
  Version line (v=) required
  Owner line (o=) required
  Session name line (s=) required
  Timespec line (t=) required
  Media supported: audio image
  Network types supported: IN
  Address types supported: IP4
  Transport types supported: RTP/AVP udptl

The following sample output shows that the time interval between consecutive NOTIFY messages for a telephone event is the default of 2000 ms:

**Example:**

```
Device# show sip-ua status
SIP User Agent Status
SIP User Agent for UDP : ENABLED
SIP User Agent for TCP : ENABLED
SIP User Agent bind status(signaling): DISABLED
SIP User Agent bind status(media): DISABLED
SIP early-media for 180 responses with SDP: ENABLED
SIP max-forwards : 6
SIP DNS SRV version: 2 (rfc 2782)
NAT Settings for the SIP-UA
Role in SDP: NONE
Check media source packets: DISABLED
Maximum duration for a telephone-event in NOTIFYs: 2000 ms
SIP support for ISDN SUSPEND/RESUME: ENABLED
Redirection (3xx) message handling: ENABLED
```
SDP application configuration:
Version line (v=) required
Owner line (o=) required
Timespec line (t=) required
Media supported: audio image
Network types supported: IN
Address types supported: IP4
Transport types supported: RTP/AVP udptl

The following sample output shows configuration of the SIP INFO Method for DTMF Tone Generation feature:

Example:

Device# show sip-ua status
SIP User Agent Status
SIP User Agent for UDP : ENABLED
SIP User Agent for TCP : ENABLED
SIP User Agent bind status(signaling): DISABLED
SIP User Agent bind status(media): DISABLED
SIP max-forwards : 6
SIP DNS SRV version: 2 (rfc 2782)
SDP application configuration:
Version line (v=) required
Owner line (o=) required
Session name line (s=) required
Timespec line (t=) required
Media supported: audio image
Network types supported: IN
Address types supported: IP4
Transport types supported: RTP/AVP udptl

Step 5 show sip-ua timers
Use this command to display the current settings for SIP user-agent timers.

Example:

Device# show sip-ua timers
SIP UA Timer Values (milliseconds)
trying 500, expires 300000, connect 500, disconnect 500
comet 500, prack 500, rel1xx 500, notify 500

Step 6 show voip rtp connections
Use this command to show local and remote Calling ID and IP address and port information.

Step 7 show sip-ua calls
Use this command to ensure the DTMF method is SIP-KPML.

The following sample output shows that the DTMF method is SIP-KPML.

Example:

Device# show sip-ua calls
SIP Call ID : 57633F68-2BE011D6-8013D46B-B4F9B5F6@172.18.193.251
State of the call : STATE_ACTIVE (7)
Substate of the call : SUBSTATE_NONE (0)
Calling Number : 
Called Number : 8888
Bit Flags : 0xD44018 0x100 0x0
CC Call ID : 6
Source IP Address (Sig ) : 192.0.0.1
Destn SIP Req Addr:Port : 192.0.2.2:5060
Destn SIP Resp Addr:Port: 192.0.2.3:5060
Destination Name : 192.0.2.4.250
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 : 6
  Stream Type : voice-only (0)
  Negotiated Codec : g711ulaw (160 bytes)
  Codec Payload Type : 0
  Negotiated Dtmf-relay : sip-kpml
  Dtmf-relay Payload Type : 0
Media Source IP Addr:Port: 192.0.2.5:17576
Media Dest IP Addr:Port : 192.0.2.6:17468
Orig Media Dest IP Addr:Port : 0.0.0.0:0
Number of SIP User Agent Client(UAC) calls: 1
SIP UAS CALL INFO
  Number of SIP User Agent Server(UAS) calls: 0

Troubleshooting Tips

- To enable debugging for RTP named-event packets, use the `debug voip rtp` command.
- To enable KPML debugs, use the `debug kpml` command.
- To enable SIP debugs, use the `debug ccsip` command.
- Collect debugs while the call is being established and during digit presses.
- If an established call is not sending digits through KPML, use the `show sip-ua calls` command to ensure SIP-KPML is included in the negotiation process.

Feature Information for DTMF Events through SIP Signaling

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 . An account on Cisco.com is not required.
### Table 4: Feature Information for Configuring DTMF Events through SIP Signaling

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTMF Events through SIP Signaling</td>
<td>12.2(11)T 12.2(8)Y 12.2(15)T 12.2(11)YV 12.2(11)T,</td>
<td>The DTMF Events through SIP Signaling feature provides the following:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• DTMF event notification for SIP messages.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Capability of receiving hookflash event notification through the SIP NOTIFY method.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Third-party call control, or other signaling mechanisms, to provide enhanced services, such as calling card and messaging services.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Communication with the application outside of the media connection.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following commands were introduced or modified: <strong>timers notify and retry notify.</strong></td>
</tr>
<tr>
<td>DTMF Events through SIP Signaling</td>
<td>Cisco IOS XE Release 2.5</td>
<td>The DTMF Events through SIP Signaling feature provides the following:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• DTMF event notification for SIP messages.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Capability of receiving hookflash event notification through the SIP NOTIFY method.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Third-party call control, or other signaling mechanisms, to provide enhanced services, such as calling card and messaging services.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Communication with the application outside of the media connection.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following commands were introduced or modified: <strong>timers notify and retry notify.</strong></td>
</tr>
</tbody>
</table>
Call Progress Analysis Over IP-to-IP Media Session

The Call Progress Analysis Over IP-IP Media Session feature enables the detection of automated answering systems and live human voices on outbound calls and communicates the detected information to the external application. Typically, call progress analysis (CPA) is extensively used in contact center deployments in conjunction with the outbound Session Initiation Protocol (SIP) dialer, where CPA is enabled on the Cisco Unified Border Element (Cisco UBE), and digital signal processors (DSP) perform the CPA functionality.

- Feature Information for Call Progress Analysis Over IP-IP Media Session, page 45
- Restrictions for Call Progress Analysis Over IP-to-IP Media Session, page 46
- Information About Call Progress Analysis Over IP-IP Media Session, page 47
- How to Configure Call Progress Analysis Over IP-to-IP Media Session, page 48
- Configuration Examples for the Call Progress Analysis Over IP-to-IP Media Session, page 51

Feature Information for Call Progress Analysis Over IP-IP Media Session

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 . An account on Cisco.com is not required.
Table 5: Feature Information for Call Progress Analysis Over IP-IP Media Session

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
</table>
| Call Progress Analysis Over IP-to-IP Media Session | 15.3(2)T                  | The Call Progress Analysis Over IP-to-IP Media Session feature enables detection of automated answering systems and live human voices on outbound calls and communicates the detected information to an external application.  
  The following command was introduced: call-progress-analysis. |
| Call Progress Analysis Over IP-to-IP Media Session | Cisco IOS XE Release 3.9S  | The Call Progress Analysis Over IP-to-IP Media Session feature enables detection of automated answering systems and live human voices on outbound calls and communicates the detected information to an external application.  
  The following command was introduced: call-progress-analysis. |
| Support for additional call flows                 | 15.5(2)T                  | Call Progress Analysis feature is enhanced to support the following call-flows:  
  • 180 SIP response received without SDP  
  • Direct call connect (without 18x from Service Provider)  
  • Multiple 18x response to INVITE  
  • Early dialog UPDATE  
  • Dialer-CUBE CPA call record |

Restrictions for Call Progress Analysis Over IP-to-IP Media Session

• Only SIP-to-SIP Early Offer (EO-to-EO) call flows are supported.
• Session Description Protocol (SDP) passthrough and flow-around media calls are not supported.
• Only the G711 flavor of codec is supported.
• High Availability (HA) is not supported.
• Skinny Client Control Protocol (SCCP)-based digital signal processor (DSP) farm is not supported.
Call Progress Analysis

Call progress analysis (CPA) is a DSP algorithm that analyzes the Real-Time Transport Protocol (RTP) voice stream to look for special information tones (SIT), fax or modem tones, human speech, and answering machine tones. CPA also passes the voice information to Cisco IOS or Cisco Unified Border Element (Cisco UBE).

CPA is initiated on receiving a new SIP INVITE with x-cisco-cpa content. While a call is in progress, the DSP or the Xcoder analyzes the incoming voice or media stream. The DSP identifies the type of voice stream based on statistical voice patterns or specific tone frequencies and provides the information to the Cisco UBE. The Cisco UBE notifies the dialer with a SIP UPDATE with x-cisco-cpa content along with the detected event. Based on the report, the caller (dialer) can decide to either transfer the call or terminate the call.

To use the CPA functionality, you must enable CPA and configure CPA timing and threshold parameters.

Table 6: X-cisco-cpa content meaning

<table>
<thead>
<tr>
<th>SIP Message</th>
<th>Direction of Message</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>18x or 200</td>
<td>Cisco IOS to dialer</td>
<td>Cisco UBE informs the dialer if CPA is enabled for a call or not.</td>
</tr>
<tr>
<td>New INVITE</td>
<td>Dialer to Cisco IOS</td>
<td>Dialer requests Cisco IOS or the Cisco UBE to activate the CPA algorithm for this session.</td>
</tr>
<tr>
<td>UPDATE</td>
<td>Cisco IOS to dialer</td>
<td>Cisco IOS or the Cisco UBE notifies the dialer about the detected event.</td>
</tr>
</tbody>
</table>

CPA Events

Table 7: CPA Event Detection List

<table>
<thead>
<tr>
<th>CPA Event</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asm</td>
<td>Answer machine</td>
</tr>
</tbody>
</table>
**How to Configure Call Progress Analysis Over IP-to-IP Media Session**

**Enabling CPA and Setting the CPA Parameters**

Perform the following task to enable CPA and set the CPA timing and threshold parameters:

**SUMMARY STEPS**

1. enable
2. configure terminal
3. dspfarm profile profile-identifier transcode
4. call-progress-analysis
5. exit
6. voice service voip
7. cpa timing live-person max-duration
8. cpa timing term-tone max-duration
9. cpa threshold active-signal signal-threshold
10. end

---

<table>
<thead>
<tr>
<th>CPA Event</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AsmT</td>
<td>Answer machine terminate tone</td>
</tr>
<tr>
<td>CpaS</td>
<td>Start of the Call Progress Analysis</td>
</tr>
<tr>
<td>FT</td>
<td>Fax/Modem tone</td>
</tr>
<tr>
<td>LS</td>
<td>Live human speech</td>
</tr>
<tr>
<td>LV</td>
<td>Low volume or dead air call</td>
</tr>
<tr>
<td>SitIC</td>
<td>Special information tone IC -- Intercept -- Vacant number or Automatic Identification System (AIS)</td>
</tr>
<tr>
<td>SitNC</td>
<td>SIT tone NC—No Circuit (NC), Emergency, or Trunk Blockage</td>
</tr>
<tr>
<td>SitVC</td>
<td>SIT tone VC—Vacant Code</td>
</tr>
<tr>
<td>SitRO</td>
<td>SIT tone RO—Reorder Announcement</td>
</tr>
<tr>
<td>SitMT</td>
<td>Miscellaneous SIT Tone</td>
</tr>
</tbody>
</table>
# DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td>• Enter your password if prompted.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> dspfarm profile  profile-identifier transcode</td>
<td>Enters DSP farm profile configuration mode, defines a profile for DSP farm services, and enables the profile for transcoding.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device(config)# dspfarm profile 15 transcode</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> call-progress-analysis</td>
<td>Enables call progress analysis (CPA) on Cisco UBE.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device(config-dspfarm-profile)# call-progress-analysis</td>
<td>You must configure this command to activate the CPA feature and set CPA parameters.</td>
</tr>
<tr>
<td><strong>Step 5</strong> exit</td>
<td>Exits DSP farm profile configuration mode and enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device(config-dspfarm-profile)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> voice service voip</td>
<td>Enters voice service configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device(config)# voice service voip</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> cpa timing live-person max-duration</td>
<td>(Optional) Sets the maximum waiting time (in milliseconds) that the CPA algorithm uses to determine if a call is answered by a live human.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device(conf-voi-serv)# cpa timing live-person 2501</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> cpa timing term-tone max-duration</td>
<td>(Optional) Sets the maximum waiting time (in milliseconds) that the CPA algorithm uses to wait for the answering machine termination tone after the answering machine is detected.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device(conf-voi-serv)# cpa timing term-tone 15500</td>
<td></td>
</tr>
</tbody>
</table>
### Purpose

Command or Action | Purpose |
--- | --- |
**Step 9** | cpa threshold active-signal signal-threshold |
**Example:**
Device(conf-voi-serv)# cpa threshold active-signal 18db |
(Optional) Sets the threshold (in decibels) of an active signal that is related to the measured noise floor level.  
• If a signal threshold configured by this command is greater than the measured noise floor level, then the signal is considered as active. The active signal thresholds that you can configure are 9, 12, 15, 18, and 21 decibels. |
**Step 10** | end |
**Example:**
Device(conf-voi-serv)# end |
Exits voice service configuration mode and returns to privileged EXEC mode. |

---

### Verifying the Call Progress Analysis Over IP-to-IP Media Session

Perform this task to verify that call progress analysis has been configured for a digital signal processor (DSP) farm profile.

**SUMMARY STEPS**

1. enable  
2. show dspfarm profile profile-identifier

**DETAILED STEPS**

**Step 1**

enable  
Enables privileged EXEC mode.  
**Example:**
Device> enable

**Step 2**

show dspfarm profile profile-identifier  
Displays the configured DSP farm profile information for a selected Cisco Call Manager group. In the following sample output, the Call Progress Analysis field shows that CPA is enabled.

**Example:**
Device# show dspfarm profile 3  
Profile ID = 3, Service -Universal TRANSCODING, Resource ID = 3  
Profile Description :  
Profile Service Mode : Non Secure  
Profile Admin State : UP  
Profile Operation State : ACTIVE
Troubleshooting Tips

Use the following commands to troubleshoot the call progress analysis for SIP-to-SIP calls:

- `debug ccsp all`
- `debug voip ecapi inout`
- `debug voip hpi all`
- `debug voip ipipgw`
- `debug voip media resource provisioning all`

Configuration Examples for the Call Progress Analysis Over IP-to-IP Media Session

Example: Enabling CPA and Setting the CPA Parameters

The following example shows how to enable CPA and set a few timing and threshold parameters. Depending on your requirements, you can configure more timing and threshold parameters.

```
Device> enable
Device# configure terminal
Device(config)# dspfarm profile 15 transcode
Device(config-dspfarm-profile)# call-progress-analysis
Device(config-dspfarm-profile)# exit
Device(config)# voice service voip
Device(config-voi-serv)# cpas timing live-person 2501
Device(config-voi-serv)# cpas timing term-tone 15500
Device(config-voi-serv)# cpas threshold active-signal 18db
Device(config-voi-serv)# end
```
Example: Enabling CPA and Setting the CPA Parameters
Codec Preference Lists

This chapter describes how to negotiate an audio codec from a list of codec associated with a preference. This chapter also describes how to disable codec filtering by configuring CUBE to send an outgoing offer with all configured audio codecs in the list assuming that the dspfarm supports all these codecs.

- Feature Information for Negotiation of an Audio Codec from a List of Codecs, page 53
- Codecs configured using Preference Lists, page 54
- Prerequisites for Codec Preference Lists, page 55
- Restrictions for Codecs Preference Lists, page 55
- How to Configure Codec Preference Lists, page 56
- Troubleshooting Negotiation of an Audio Codec from a List of Codecs, page 59
- Verifying Negotiation of an Audio Codec from a List of Codecs, page 59

Feature Information for Negotiation of an Audio Codec from a List of Codecs

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. An account on Cisco.com is not required.
Table 8: Feature Information for Negotiation of an Audio Codec from a List of Codecs on Each Leg of a SIP-to-SIP Call on the Cisco Unified Border Element

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Negotiation of an Audio Codec from a List of Codecs on Each Leg of a SIP-to-SIP Call on the Cisco Unified Border Element</td>
<td>15.1(2)T</td>
<td>The Negotiation of an Audio Codec from a List of Codecs on Each Leg of a SIP-to-SIP Call on the Cisco Unified Border Element feature supports negotiation of an audio codec using the Voice Class Codec and Codec Transparent infrastructure on the Cisco UBE. The following command was introduced or modified: <code>voice-class codec (dial peer)</code>.</td>
</tr>
<tr>
<td>Negotiation of an Audio Codec from a List of Codecs on Each Leg of a SIP-to-SIP Call on the Cisco Unified Border Element</td>
<td>Cisco IOS XE Release 3.8S</td>
<td>The Negotiation of an Audio Codec from a List of Codecs on Each Leg of a SIP-to-SIP Call on the Cisco Unified Border Element feature supports negotiation of an audio codec using the Voice Class Codec and Codec Transparent infrastructure on the Cisco UBE. The following command was introduced or modified: <code>voice-class codec (dial peer)</code>.</td>
</tr>
<tr>
<td>Negotiation of an Audio Codec from a List of Codecs on Each Leg of a SIP-to-SIP Call on the Cisco Unified Border Element</td>
<td>15.3(2)T</td>
<td>This feature provides high availability support for negotiation of an audio codec from a list of codecs on each leg of a SIP-to-SIP call on the Cisco Unified Border Element under the Voice Class Codec.</td>
</tr>
</tbody>
</table>

**Codecs configured using Preference Lists**

SIP-to-SIP calls configured using codecs using preference lists have the following features:

- Incoming and outgoing dial-peers can be configured with different preference lists.
- Both normal transcoding and high-density transcoding are supported with preference lists.
- Mid-call codec changes for supplementary services are supported with preference lists. Transcoder resources are dynamically inserted or deleted when there is a codec or RTP-NTE to inband DTMF interworking required.
• Reinvite-based supplementary services invoked from the Cisco Unified Communications Manager (CUCM), like call hold, call resume, music on hold (MOH), call transfer, and call forward are supported with preference lists.

• T.38 fax and fax passthrough switchover with preference lists are supported.

• Reinvite-based call hold and call resume for Secure Real-Time Transfer protocol (SRTP) and Real-Time Transport Protocol (RTP) interworking on CUBE is supported with preference lists.

• High availability is supported for calls that use codecs with preference lists. But calls requiring the transcoder to be invoked are not checkpointed. During mid-call renegotiation, if the call releases the transcoder, then the call is checkpointed.

Prerequisites for Codec Preference Lists

• Transcoding configuration on the CUBE.

• The digital signal processor (DSP) requirements to support the transcoding feature on the CUBE.

Restrictions for Codecs Preference Lists

For All Calls (SIP-to-SIP, H323-to-H323, SIP-to-H323 calls)

• Video codecs are not supported with preference lists.

• Multiple audio streams are not supported.

• High-density transcoding is not supported when delayed offer to early offer is configured. Only low density transcoding is supported.

• Codec re-packetization feature is not supported when preference lists are configured.

For H323-to-H323 and SIP-to-H323 Calls

The below restrictions do not exist for SIP-to-SIP calls from 15.1(2)T and Cisco IOS XE Release 3.8S onwards.

• You can configure dissimilar preference lists on the incoming and outgoing dial peers.

• Incoming and outgoing dial-peers cannot be configured with the different preference lists.

• Transcoding is not supported when preference lists are used.

• Mid-call codec changes and supplementary services (call-hold / resume, call forward) do not work when a preference list is configured.

• Mid-call insertion or deletion of transcoder is not supported with preference lists.

• Rotary dial peers are not supported when preference lists are used.

• Both incoming and outgoing dial-peers need to be configured with the same codec voice classes.

• The preference of codecs configured in a codec voice classes is not be applied to the outgoing call-leg. Basically codec filtering is applied first and only the filtered codecs will be sent out in the outgoing offer from CUBE.
How to Configure Codec Preference Lists

Configuring Audio Codecs Using a Codec Voice Class and Preference Lists

Preferences can be used to determine which codecs will be selected over others.

A codec voice class is a construct within which a codec preference order can be defined. A codec voice class can then be applied to a dial peer, which then follows the preference order defined in the codec voice class.

SUMMARY STEPS

1. enable
2. configure terminal
3. voice class codec tag
4. Do the following for each audio codec you want to configure in the voice class:
   - codec preference value codec-type [bytes payload-size fixed-bytes ]
   - codec preference value isac [mode {adaptive | independent} [bit-rate value framesize { 30 | 60 } [fixed] ]]
   - codec preference value ilbc [mode frame-size [bytes payload-size]]
   - codec preference value mp4-latm [profile tag]
5. exit
6. dial-peer voice number voip
7. voice-class codec tag offer-all
8. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td></td>
</tr>
<tr>
<td>enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td></td>
</tr>
<tr>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device&gt; configure terminal</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>voice class codec <em>tag</em></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config)# voice class codec 10</td>
</tr>
</tbody>
</table>
| **Step 4** | Do the following for each audio codec you want to configure in the voice class:  
  - codec preference *value* codec-type[*bytes* payload-size *fixed-bytes*]  
  - codec preference *value* isac [mode {adaptive | independent} [bit-rate *value* framesize {30 | 60} [fixed]]  
  - codec preference *value* ilbc [mode frame-size [bytes payload-size]]  
  - codec preference *value* mp4-latm [profile *tag*] | Configure a codec within the voice class and specifies a preference for the codec. This becomes part of a preference list |
| **Step 5** | exit | Exits the current mode.  
  - Enter your password if prompted. |
| **Example:** | Device(config-class)# exit |
| **Step 6** | dial-peer voice *number* voip | Enters dial peer configuration mode for the specified VoIP dial peer. |
| **Example:** | Device(config)# dial-peer voice 1 voip |
| **Step 7** | voice-class codec *tag* offer-all | Applies the previously configured voice class and associated codecs to a dial peer.  
  - The *offer-all* keyword allows the device to offer all codecs configured in a codec voice class. |
| **Example:** | Device(config-dial-peer)# voice-class codec 10 |
| **Step 8** | end | Returns to privileged EXEC mode. |
| **Example:** | Device(config-dial-peer)# end |

### Disabling Codec Filtering

Cisco UBE is configured to filter common codecs for the subsets, by default. The filtered codecs are sent in the outgoing offer. You can configure the Cisco UBE to offer all the codecs configured on an outbound leg instead of offering only the filtered codecs.
This configuration is applicable only for early offer calls from the Cisco UBE. For delayed offer calls, by default all codecs are offered irrespective of this configuration.

Perform this task to disable codec filtering and allow all the codecs configured on an outbound leg.

**SUMMARY STEPS**

1. enable
2. configure terminal
3. dial-peer voice \( tag \) voip
4. voice-class codec \( tag \) offer-all
5. end

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td></td>
<td>Example: Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example: Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>dial-peer voice ( tag ) voip</td>
<td>Enters dial peer voice configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example: Device(config)# dial-peer voice 10 voip</td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>voice-class codec ( tag ) offer-all</td>
<td>Adds all the configured voice class codec to the outgoing offer from the Cisco UBE.</td>
</tr>
<tr>
<td></td>
<td>Example: Device(config-dial-peer)# voice-class codec 10 offer-all</td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td>end</td>
<td>Exits the dial peer voice configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example: Device(config-dial-peer)# end</td>
<td></td>
</tr>
</tbody>
</table>
Troubleshooting Negotiation of an Audio Codec from a List of Codecs

Use the following commands to debug any errors that you may encounter when you configure the Negotiation of an Audio Codec from a List of Codecs on Each Leg of a SIP-to-SIP Call on the Cisco Unified Border Element feature:

- `debug ccsip all`
- `debug voip ccapi input`
- `debug sccp messages`
- `debug voip rtp session`

For DSP-related debugs, use the following commands:

- `debug voip dsmp all`
- `debug voip dsmp rtp both payload all`
- `debug voip ipipgw`

Verifying Negotiation of an Audio Codec from a List of Codecs

Perform this task to display information to verify Negotiation of an Audio Codec from a List of Codecs on Each Leg of a SIP-to-SIP Call on the Cisco Unified Border Element configuration. These `show` commands need not be entered in any specific order.

**SUMMARY STEPS**

1. `enable`
2. `show call active voice brief`
3. `show voip rtp connections`
4. `show sccp connections`
5. `show dspfarm dsp active`

**DETAILED STEPS**

**Step 1**

`enable`

Enables privileged EXEC mode.

**Step 2**

`show call active voice brief`

Displays a truncated version of call information for voice calls in progress.

**Example:**

```
Device# show call active voice brief
```
Verifying Negotiation of an Audio Codec from a List of Codecs

### Codec Preference Lists

#### Step 3

**show voip rtp connections**

Displays Real-Time Transport Protocol (RTP) connections.

**Example:**

```
Device# show voip rtp connections
VoIP RTP active connections :
  No. CallId dstCallId LocalRTP RmtRTP LocalIP RemoteIP
  1 11 12 16662 19304 192.0.2.1
  2 12 11 17404 16512 192.0.2.2
  3 13 14 18422 2000 9.44.26.3
  4 15 14 16576 2000 192.0.2.6
```

---

Cisco Unified Border Element Protocol-Independent Features and Setup Configuration Guide, Cisco IOS Release 15M&T

---
192.0.2.5
Found 4 active RTP connections

Step 4  **show sccp connections**
Displays information about the connections controlled by the Skinny Client Control Protocol (SCCP) transcoding and conferencing applications.

Example:

```
Device# show sccp connections
sess_id  conn_id  stype  mode  codec  sport  rport  ripaddr
5        5        xcode  sendrecv  g729b  16576  2000  192.0.2.3
5        6        xcode  sendrecv  g711u  18422  2000  192.0.2.4
Total number of active session(s) 1, and connection(s) 2
```

Step 5  **show dspfarm dsp active**
Displays active DSP information about the DSP farm service.

Example:

```
Device# show dspfarm dsp active
SLOT  DSP VERSION  STATUS  CHNL  USE  TYPE  RSC_ID  BRIDGE_ID  PKTS_TXED  PKTS_RXED
0     1    27.0.201  UP      1    USED  xcode  1    0x9       5    8
0     1    27.0.201  UP      1    USED  xcode  1    0x8      2558   17
Total number of DSPPARM DSP channel(s) 1
```
Verifying Negotiation of an Audio Codec from a List of Codecs
CHAPTER 8

AAC-LD MP4A-LATM Codec Support on Cisco UBE

The AAC-LD MP4A-LATM codec is a wideband audio codec used by video endpoints. MP4A-LATM is an MPEG4 audio coding standard, where LATM is Low-Overhead MPEG-4 Audio Transport Multiplex. The Cisco Unified Border Element (Cisco UBE) supports MP4A-LATM to enable call flows involving endpoints that use this codec, especially for media recording.

For basic information on Codecs and how to configure them, refer to Codecs in the Cisco Unified Border Element Fundamentals and Basic Setup.

- Finding Feature Information, page 63
- Restrictions for AAC-LD MP4A-LATM Codec Support on Cisco UBE, page 64
- AAC-LD MP4A-LATM Codec Support on Cisco UBE, page 64
- How to Configure the MP4A-LATM Codec, page 65
- Configuration Examples for AAC-LD MP4A-LATM Codec Support on Cisco UBE, page 69
- Feature Information for AAC-LD MP4A-LATM Codec Support on Cisco UBE, page 70

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.
Restrictions for AAC-LD MP4A-LATM Codec Support on Cisco UBE

Cisco UBE does not support the following:

• Codec transcoding between MP4A-LATM and other codecs
• Dual-tone Multifrequency (DTMF) interworking with MP4A-LATM codec
• Non-SIP-SIP, that is, SIP to other service provider interface (SPI) interworking with MP4A-LATM codec

AAC-LD MP4A-LATM Codec Support on Cisco UBE

As part of this feature, Cisco UBE supports the following:

• Accept and send MP4A-LATM codec and corresponding FMTP profiles
• Configure MP4A-LATM under dial-peer or under voice-class codec as preferred codec
• Pass across real-time transport protocol (RTP) media for MP4A-LATM codec without any interworking
• Offer pre-configured FMTP profile for MP4A-LATM for DO-EO (Delayed-Offerto Early-Offer) calls
• Offer more than one FMTP profile (each with different payload type number) as mentioned by the offering endpoint, so that the answering endpoint can choose the best option.
• Offer only one instance of MP4A-LATM if media forking is applicable. The offered instance is the first one received in the offer.
• Calculate bandwidth for MP4A-LATM on the basis of either “b=TIAS” attribute or “bitrate” parameter in the FMTP attribute. If none of them are present in the session description protocol (SDP), the default maximum bandwidth, that is, 128 Kbps will be used for calculation.
• The following Cisco UBE features are supported with the MP4A-LATM codec:
  ◦ Basic call (audio and video) flow-around and flow-through (FA and FT).
  ◦ Voice Class Codec support in Cisco UBE with codec filtering
  ◦ SRTP and SRCTP passthrough for SIP-to-SIP calls
  ◦ Supplementary services
  ◦ RSVP
  ◦ Dynamic payload type interworking for DTMF and codec packets for SIP-to-SIP calls
  ◦ Media Anti-Trombone with SIP signaling control on CUBE
  ◦ Support for SIP UPDATE message per RFC 3311
  ◦ RTP Media Loopback
  ◦ Media forking for IP based calls using Zephyr recording server
  ◦ Cisco UBE Mid-call Re-INVITE consumption
How to Configure the MP4A-LATM Codec

Configuring the MP4A-LATM Codec on a Dial Peer

**SUMMARY STEPS**

1. `enable`
2. `configure terminal`
3. `dial-peer voice tag voip`
4. `destination-pattern [+] string [T]`
5. `session protocol sipv2`
6. `session target ipv4:destination-address`
7. `codec mp4a-latm [profile tag]`
8. `end`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enters privileged EXEC mode or any other security level set by a system administrator. Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> dial-peer voice tag voip</td>
<td>Specifies the method of voice encapsulation and enters dial peer voice configuration mode for the specified dial peer.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config)# dial-peer voice 24 voip</td>
<td></td>
</tr>
</tbody>
</table>
### Configuring the MP4A-LATM Codec on a Dial Peer

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 4</strong></td>
<td><strong>destination-pattern</strong> [+]<em>string</em> [<em>T</em>]</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config-dial-peer)# destination-pattern 595959</td>
</tr>
<tr>
<td>Specifies either the prefix or the full E.164 telephone number (depending on your dial plan) to be used for a dial peer. Keywords and arguments are as follows:</td>
<td></td>
</tr>
<tr>
<td>• + --(Optional) Character that indicates an E.164 standard number.</td>
<td></td>
</tr>
<tr>
<td>• string --Series of digits that specify the E.164 or private dialing plan telephone number. Valid entries are the digits 0 through 9, the letters A through D, and any special character.</td>
<td></td>
</tr>
<tr>
<td>• T --(Optional) Control character indicating that the destination-pattern value is a variable-length dial string.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td><strong>session protocol sipv2</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config-dial-peer)# session protocol sipv2</td>
</tr>
<tr>
<td>Configures the VoIP dial peer to use Session Initiation Protocol (SIP).</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>**session target ipv4:**destination-address</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config-dial-peer)# session target ipv4:10.42.29.7</td>
</tr>
<tr>
<td>Specifies a network-specific address for a dial peer. Keyword and argument are as follows:</td>
<td></td>
</tr>
<tr>
<td>• ipv4: destination address --IP address of the dial peer, in this format: xxx.xxx.xxx.xxx</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td><strong>codec mp4a-latm</strong> [profile <em>tag</em>]</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config-dial-peer)# codec mp4a-latm profile 5</td>
</tr>
<tr>
<td>Configures the MP4A-LATM codec for the dial peer.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td><strong>end</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config-dial-peer)# end</td>
</tr>
<tr>
<td>Exits dial peer voice configuration mode.</td>
<td></td>
</tr>
</tbody>
</table>
### Configuring the MP4A-LATM Codec under Voice Class Codec

#### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice class codec tag**
4. **codec preference value codec-type [profile tag]**
5. **exit**
6. **dial-peer voice tag voip**
7. **destination-pattern [+ string] [T]**
8. **session protocol sipv2**
9. **session target ipv4:destination-address**
10. **voice-class codec tag**

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters privileged EXEC mode or any other security level set by a system administrator. Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>enable</strong></td>
<td>Example: Device&gt; enable</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>configure terminal</strong></td>
<td>Example: Device# configure terminal</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Enters voice-class configuration mode and assigns an identification tag number for a codec voice class.</td>
</tr>
<tr>
<td><strong>voice class codec tag</strong></td>
<td>Example: Device(config)# voice class codec 1</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Specifies the preferred codec (or codecs) to use on a dial peer.</td>
</tr>
<tr>
<td><strong>codec preference value codec-type [profile tag]</strong></td>
<td>Example: Device(config-class)# codec preference 1 mp4a-latm profile 5</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Exits voice-class configuration mode.</td>
</tr>
<tr>
<td><strong>exit</strong></td>
<td>Example: Device(config-class)# exit</td>
</tr>
<tr>
<td>Step 6</td>
<td>Command or Action</td>
</tr>
<tr>
<td>--------</td>
<td>-------------------</td>
</tr>
<tr>
<td></td>
<td>dial-peer voice tag voip</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config)# dial-peer voice 24 voip</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 7</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>destination-pattern [+ string [T]]</td>
<td>Specifies either the prefix or the full E.164 telephone number (depending on your dial plan) to be used for a dial peer. Keywords and arguments are as follows:</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config-dial-peer)# destination-pattern 595959</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 8</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>session protocol sipv2</td>
<td>Configures the VoIP dial peer to use Session Initiation Protocol (SIP).</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config-dial-peer)# session protocol sipv2</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 9</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>session target ipv4:destination-address</td>
<td>Specifies a network-specific address for a dial peer. Keyword and argument are as follows:</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config-dial-peer)# session target ipv4:10.42.29.7</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 10</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>voice-class codec tag</td>
<td>Enters voice-class configuration mode and assigns an identification tag number for a codec voice class.</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config-dial-peer)# voice-class codec 1</td>
<td></td>
</tr>
</tbody>
</table>

### Verifying an Audio Call

**SUMMARY STEPS**

1. `show call active voice [compact]`
DETAILED STEPS

show call active voice [compact]
Displays a compact version of call information for voice calls in progress.

Example:
Device# show call active voice compact

<table>
<thead>
<tr>
<th>&lt;callID&gt;</th>
<th>A/O FAX T&lt;sec&gt; Codec</th>
<th>type</th>
<th>Peer Address</th>
<th>IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>23 ANS</td>
<td>T3</td>
<td>mp4a-latm</td>
<td>VOIP</td>
<td>Psipp</td>
</tr>
<tr>
<td>24 ORG</td>
<td>T3</td>
<td>mp4a-latm</td>
<td>VOIP</td>
<td>P123</td>
</tr>
</tbody>
</table>

Example:
Device# show call active voice compact

<table>
<thead>
<tr>
<th>&lt;callID&gt;</th>
<th>A/O FAX T&lt;sec&gt; Codec</th>
<th>type</th>
<th>Peer Address</th>
<th>IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>58 ANS</td>
<td>T11</td>
<td>g711ulaw</td>
<td>VOIP</td>
<td>Psipp 2001:......:230A:6080</td>
</tr>
<tr>
<td>59 ORG</td>
<td>T11</td>
<td>g711ulaw</td>
<td>VOIP</td>
<td>P5000110011</td>
</tr>
</tbody>
</table>

Configuration Examples for AAC-LD MP4A-LATM Codec Support on Cisco UBE

Example: Configuring the MP4A-LATM Codec under a Dial Peer

Device> enable
Device# configure terminal
Device(config)# dial-peer voice 24 voip
Device(config)# destination-pattern 595959
Device(config)# session protocol sipv2
Device(config)# session target ipv4:10.42.29.7
Device(config)# codec mp4a-latm profile 5
Device(config)# end

Example: Configuring the MP4A-LATM Codec under Voice Class Codec

Device> enable
Device# configure terminal
Device(config)# voice class codec 1
Device(config)# codec preference 1 mp4a-latm profile 5
Device(config)# exit
Device(config)# dial-peer voice 24 voip
Device(config)# destination-pattern 595959
Device(config)# session protocol sipv2
Device(config)# session target ipv4:10.42.29.7
Device(config)# voice-class codec 1
Feature Information for AAC-LD MP4A-LATM Codec Support on Cisco UBE

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 . An account on Cisco.com is not required.

Table 9: Feature Information for AAC-LD MP4A-LATM Codec Support on Cisco UBE

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAC-LD MP4A-LATM Codec Support on Cisco UBE</td>
<td>15.4(1)T</td>
<td>The AAC-LD MP4A-LATM codec is a wideband audio codec used by video endpoints. MP4A-LATM is an MPEG4 audio coding standard, where LATM is Low-Overhead MPEG-4 Audio Transport Multiplex. The Cisco Unified Border Element (Cisco UBE) supports MP4A-LATM to enable call flows involving endpoints that use this codec, especially for media recording. The following commands were introduced or modified: <code>codec mp4a-latm</code>, <code>codec preference tag mp4a-latm</code></td>
</tr>
</tbody>
</table>
Multicast Music-on-Hold Support on Cisco UBE

First Published: July 22, 2011
Last Updated: July 22, 2011

The Multicast Music-on-Hold (MMOH) feature enables you to subscribe to a music streaming service when you are using a Cisco Unified Border Element. Music streams from an MMOH server to the interface of Cisco UBE, which then converts it into unicast. To play the MMOH to customers using Cisco UBE, you must enable the MMOH feature on Cisco UBE.

• Prerequisites for Multicast Music-on-Hold Support on Cisco UBE, page 71
• Restrictions for Multicast Music-on-Hold Support on Cisco UBE, page 71
• Information About Multicast Music-on-Hold Support on Cisco UBE, page 72
• How to Enable Multicast Music-on-Hold on Cisco UBE, page 72
• Configuration Examples for Multicast Music-on-Hold Support on Cisco UBE, page 77
• Feature Information for Multicast Music-on-Hold Support on Cisco UBE, page 79

Prerequisites for Multicast Music-on-Hold Support on Cisco UBE

Cisco Unified Border Element

• Cisco IOS Release 15.2(1)T or a later release must be installed and running on your Cisco Unified Border Element.

Restrictions for Multicast Music-on-Hold Support on Cisco UBE

• The Multicast Music-on-Hold (MMOH) feature will not work when the Session Description Protocol (SDP) Passthrough feature is enabled on Cisco UBE.
• The MMOH feature will work for Low Density Transcoded calls but not for High Density Transcoded calls.
• MMOH is supported only on SIP-to-SIP call flows on Cisco UBE.
• MMOH with RTCP is not supported.
• MMOH is not supported for SRTP trunk.
• MMOH with media flow-around is not supported.

Information About Multicast Music-on-Hold Support on Cisco UBE

Multicast Music-on-Hold

To play Multicast Music-on-Hold (MMOH) to customers using Cisco UBE, you must enable the MMOH feature on Cisco UBE. When Cisco UBE receives an MMOH call, it converts the multicast address received on the inbound leg into a unicast address and sends the address on the outbound leg.

Cisco UBE uses preconfigured CLIs to "listen" for Real-Time Transport Protocol (RTP) packets that are broadcast from an MMOH server in the network and converts them to unicast. When a call is placed on hold, the MOH server streams the RTP packets to the Cisco UBE interface. This interface converts the RTP packets to unicast and relays the packets to the appropriate voice interfaces that have been placed on hold.

Note

MMOH is already supported on SIP-TDM gateways.

How to Enable Multicast Music-on-Hold on Cisco UBE

Enabling MMOH on Cisco UBE

Perform this task to enable the MMOH feature on Cisco UBE.
### SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `ip multicast-routing distributed`
4. `interface gigabitethernet router-shelf/slot/port`
5. `ip address ip-address subnet-mask`
6. `ip pim dense-mode`
7. `negotiation auto`
8. `exit`
9. `ccm-manager music-on-hold`
10. `exit`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** enable | Enables privileged EXEC mode.  
| Example: Device> enable | • Enter your password if prompted. |
| **Step 2** `configure terminal` | Enters global configuration mode.  
| Example: Device# configure terminal | |
| **Step 3** `ip multicast-routing distributed` | Enables distributed IP multicast routing.  
| Example: Device(config)# ip multicast-routing distributed | |
| **Step 4** `interface gigabitethernet router-shelf/slot/port` | Configures a Gigabit Ethernet interface and enters interface configuration mode.  
| Example: Device(config)# interface gigabitethernet 0/0/0 | |
| **Step 5** `ip address ip-address subnet-mask` | Configures the IP address and the subnet mask on the interface.  
| Example: Device(config-if)# ip address 9.40.1.140 255.255.0.0 | |
### Purpose

**Command or Action**

**Purpose**

<table>
<thead>
<tr>
<th>Step 6</th>
<th>ip pim dense-mode</th>
<th>Enables protocol-independent multicast (PIM) dense-mode operation.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Example: Device(config-if)# ip pim dense-mode</td>
<td></td>
</tr>
<tr>
<td>Step 7</td>
<td>negotiation auto</td>
<td>Performs link auto-negotiation.</td>
</tr>
<tr>
<td></td>
<td>Example: Device(config-if)# negotiation auto</td>
<td></td>
</tr>
<tr>
<td>Step 8</td>
<td>exit</td>
<td>Exits interface configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example: Device(config-if)# exit</td>
<td></td>
</tr>
<tr>
<td>Step 9</td>
<td>ccm-manager music-on-hold</td>
<td>Enables the multicast music-on-hold feature on a voice gateway.</td>
</tr>
<tr>
<td></td>
<td>Example: Device(config)# ccm-manager music-on-hold</td>
<td></td>
</tr>
<tr>
<td>Step 10</td>
<td>exit</td>
<td>Exits global configuration mode and enters privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>Example: Device(config)# exit</td>
<td></td>
</tr>
</tbody>
</table>

### Verifying the MMOH Support on Cisco UBE

Perform this task to verify the MMOH support on Cisco UBE. The `show` commands can be entered in any order.

### SUMMARY STEPS

1. enable
2. show ccm-manager music-on-hold
3. show voip rtp connections
4. show call active voice compact
5. show platform hardware qfp active feature sbc mmoh global
6. show platform hardware qfp active feature sbc mmoh group
DETAILED STEPS

Step 1  enable
Enables privileged EXEC mode.

Example:
Device> enable

Step 2  show ccm-manager music-on-hold
Displays information about all the multicast music-on-hold (MOH) sessions in the gateway at any given time.

Example:
Device# show ccm-manager music-on-hold
Current active multicast sessions: 1
Multicast Address   RTP port number   Packets in/out  CallId  Codec  Incoming Interface
239.1.1.1           16386            614/614          132     g711ulaw
Gi0/0

Step 3  show voip rtp connections
Displays RTP-named event packets.

Example:
Device# show voip rtp connections
VoIP RTP Port Usage Information:
Max Ports Available: 20000, Ports Reserved: 101, Ports in Use: 2
Port range not configured, Min: 8000, Max: 48200
Ports Ports
Media-Address Range Available Reserved In-use
Default Address-Range 20000 101 2

VoIP RTP active connections:
No. CallId dstCallId LocalRTP RmtRTP LocalIP RemoteIP
1 140 141 18792 18638 9.42.30.10 9.42.30.32
2 141 140 19256 26184 9.42.30.10 9.42.30.189

Found 2 active RTP sessions

Step 4  show call active voice compact
Displays a compact version of voice calls in progress.

Example:
Device# show call active voice compact
<callID> A/O FAX T<sec> Codec  type  Peer Address  IP R<ip>:<udp>
Total call-legs: 3
140 ANS    T644  g711ulaw  VOIP    P10000  9.42.30.32:18638
141 ORG    T644  g711ulaw  VOIP    P708090 9.42.30.189:26184
145 ORG    T643  g711ulaw  VOIP    P595959  9.42.29.7:3852

Step 5  show platform hardware qfp active feature sbc mmoh global
Displays SBC multicast Music-on-Hold global statistics.
Example:

Device# show platform hardware qfp active feature sbc mmoh global

SBC multicast Music-on-Hold Global Statistics

Total MMOH groups = 1
Total RTP packets received = 6311
Total RTP octects received = 1262200
Total RTP packets replicated = 6311
Total RTP octects replicated = 1262200
Total RTP packets dropped = 0
Total RTP octects dropped = 0

Step 6  
show platform hardware qfp active feature sbc mmoh group
Displays SBC multicast Music-on-Hold group structure.

Example:

Device# show platform hardware qfp active feature sbc mmoh group

SBC multicast Music-on-Hold group structure:

---------------------------------------
VRF = 0
IP = 239.1.1.1
Port = 16384
Protocol = 1
Calls in group = 1

SBC MMOH group Statistics

---------------------------------------
Total RTP packets received = 406
Total RTP octects received = 81200
Total RTP packets replicated = 406
Total RTP octects replicated = 81200
Total RTP packets dropped = 0
Total RTP octects dropped = 0

Troubleshooting Tips

The following commands can help troubleshoot MMOH:

- debug ccm-manager music-on-hold [ all | errors | events ]
- debug voip rtp
- debug ccsip all
Configuration Examples for Multicast Music-on-Hold Support on Cisco UBE

Example: Enabling MMOH on Cisco UBE

Device> enable
Device# configure terminal
Device(config)# ip multicast-routing distributed
Device(config)# interface gigabitethernet 0/0/0
Device(config-if)# ip address 9.40.1.140 255.255.0.0
Device(config-if)# ip pim dense-mode
Device(config-if)# negotiation auto
Device(config-if)# exit
Device(config)# ccm-manager music-on-hold
Device# show running-config
Building configuration... 
Current configuration : 2375 bytes

! Last configuration change at 11:01:36 UTC Wed Jan 5 2011

version 15.1
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname carbon-1
!
boot-start-marker
boot system flash usbflash0:c2951-universalk9-mz.SSA.MMOH-carbon_dev
boot-end-marker
!
!
no aaa new-model
!
no ipv6 cef
ip source-route
ip cef
!
!
ip multicast-routing
!
no ip domain lookup
multilink bundle-name authenticated
!
!
!
crypto pki token default removal timeout 0
!
voice-card 0
!
!
voice service voip
mode border-element license capacity 1200
allow-connections sip to sip
sip
Example: Enabling MMOH on Cisco UBE

```
license udi pid CISCO2951/K9 sn FHK1433F39H
hw-module pvdm 0/0

redundancy inter-device

redundancy

interface GigabitEthernet0/0
ip address 9.42.30.12 255.255.0.0
duplex auto
speed auto

interface GigabitEthernet0/1
no ip address
shutdown
duplex auto
speed auto

interface GigabitEthernet0/2
no ip address
shutdown
duplex auto
speed auto

ip forward-protocol nd

no ip http server
no ip http secure-server

ip route 0.0.0.0 0.0.0.0 9.42.0.1

nls resp-timeout 1
cpd cr-id 1

control-plane

ccm-manager music-on-hold

mgcp profile default

dial-peer voice 100 voip
  destination-pattern 878767
  session protocol sipv2
  session target ipv4:9.42.30.5
  codec g711ulaw

gatekeeper
  shutdown
  
line con 0
  speed 115200
line aux 0
line vty 0 4
```
Feature Information for Multicast Music-on-Hold Support on Cisco UBE

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 software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 10: Feature Information for Multicast Music-on-Hold Support on Cisco UBE

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multicast Music-on-Hold Support on Cisco UBE</td>
<td>15.2(1)T Cisco IOS XE Release 3.11S</td>
<td>The Multicast Music-on-Hold (MMOH) feature enables you to subscribe to a music streaming service when you are using a Cisco Unified Border Element. To play MMOH to customers using Cisco UBE, you must enable the MMOH feature on Cisco UBE. No new commands were introduced or modified.</td>
</tr>
</tbody>
</table>
Network-Based Recording

The Network-Based Recording feature supports software-based forking for Real-time Transport Protocol (RTP) streams. Media forking provides the ability to create midcall multiple streams (or branches) of audio and video associated with a single call and then send the streams of data to different destinations. To enable network-based recording using Cisco Unified Border Element (CUBE), you can configure specific commands or use a call agent. CUBE acts as a recording client and MediaSense Session Initiation Protocol (SIP) recorder acts as a recording server.

- Feature Information for Network-Based Recording, page 81
- Restrictions for Network-Based Recording, page 82
- Information About Network-Based Recording Using CUBE, page 83
- How to Configure Network-Based Recording, page 88
- Additional References for Network-Based Recording, page 107

Feature Information for Network-Based Recording

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 software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.
### Table 11: Feature Information for Network-Based Recording

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audio-only Stream Forking of Video Call</td>
<td>Cisco IOS 15.4(3)M Cisco IOS XE 3.13S</td>
<td>The Audio-only Stream Forking of Video Call feature supports CUBE-based forking and recording of only audio calls in a call that includes both audio and video. The following commands were introduced: media-type audio.</td>
</tr>
<tr>
<td>Network-Based Recording of Video Calls Using CUBE</td>
<td>Cisco IOS 15.3(3)M Cisco IOS XE 3.10S</td>
<td>The Network-Based Recording of Video Calls using CUBE feature supports forking and recording of video calls.</td>
</tr>
<tr>
<td>Network-Based Recording of Audio Calls Using CUBE</td>
<td>Cisco IOS 15.2(1)T Cisco IOS XE 3.8S</td>
<td>The Network-Based Recording of Audio Calls using CUBE feature supports forking for RTP streams. The following commands were introduced or modified: media class, media profile recorder, media-recording, recorder parameter, recorder profile, show voip recmsp session.</td>
</tr>
</tbody>
</table>

### Restrictions for Network-Based Recording

- Network-based recording is not supported for the following calls:
  - Calls that do not use Session Initiation Protocol (SIP). Must be a SIP-to-SIP call flow
  - Flow-around calls
  - Session Description Protocol (SDP) pass-through calls
  - Real-time Transport Protocol (RTP) loopback calls
  - High-density transcoder calls
  - IPv6-to-IPv6 calls
  - IPv6-to-IPv4 calls with IPv4 endpoint.
  - Secure Real-time Transport Protocol (SRTP) pass-through calls
  - SRTP-RTP calls with forking for SRTP leg (forking is supported for the RTP leg)
  - Resource Reservation Protocol (RSVP)
  - Multicast music on hold (MOH)
Restrictions for Video Recording

- If the main call has multiple video streams (m-lines), the video streams other than the first video m-line are not forked.
- Application media streams of the primary call are not forked to the recording server.
- Forking is not supported if the anchor leg or recording server is on IPv6.
- High availability is not supported on forked video calls.

Information About Network-Based Recording Using CUBE

Deployment Scenarios for CUBE-based Recording

CUBE as a recording client has the following functions:

- Acts as a SIP user agent and sets up a recording session (SIP dialog) with the recording server.
- Acts as the source of the recorded media and forwards the recorded media to the recording server.
- Sends information to a server that helps the recording server associate the call with media streams and identifies the participants of the call. This information sent to the recording server is called metadata.

Given below is a typical deployment scenario of a CUBE-based recording solution. The information flow is described below:

Figure 1: Deployment Scenario for CUBE-based Recording Solution
Incoming call from SIP trunk.
2 Outbound call to a Contact Centre
3 Media between endpoints flow through CUBE
4 CUBE sets up a new SIP session with MediaSense based on policy.
5 CUBE forks RTP media to MediaSense. For an audio call, audio is forked. For a video call, both audio and video are forked. For an audio-only configuration in a audio-video call, only audio is forked. There will be two or four m-lines to the recording server, based on the type of recording.

The metadata carried in the SIP session between the recording client and the recording server is to:

- Carry the communication session data that describes the call.
- Send the metadata to the recording server. The recording server uses the metadata to associate communication sessions involving two or more participants with media streams.

The call leg that is created between the recording client and the recording server is known as the recording session.

Open Recording Architecture

The Open Recording Architecture (ORA) comprises of elements, such as application management server and SIP bridge, to support IP-based recording. The ORA IP enables recording by solving topology issues, which accelerates the adoption of Cisco unified communication solutions.
Following are the three layers of the ORA architecture:

**Network Layer**

The ORA network layer is comprised of control systems, media sources, and IP foundation components, such as routers and switches.

**Capture and Media Processing Layer**

The ORA capture and media processing layer includes core functions of ORA—terminating media streams, storage of media and metadata, and speech analytics that can provide real-time events for applications.

**Application Layer**

The ORA application layer supports in-call and post-call applications through open programming interfaces. In-call applications include applications that make real-time business decisions (for example, whether to record a particular call or not), control pause and resume from Interactive Voice Response (IVR) or agent desktop systems, and perform metadata tagging and encryption key exchange at the call setup.

Post-call applications include the following:

- Traditional compliance search, replay, and quality monitoring.
- Advanced capabilities, such as speech analytics, transcription, and phonetic search.
• Custom enterprise integration.
• Enterprise-wide policy management.

Media Forking Topologies

The following topologies support media forking:

Media Forking with Cisco UCM

The figure below illustrates media forking with Cisco Unified CallManager (Cisco UCM) topology. This topology supports replication of media packets to allow recording by the caller agent. It also enables CUBE to establish full-duplex communication with the recording server. In this topology, SIP recording trunk is enhanced to have additional call metadata.

Media Forking without Cisco UCM

The topology below shows media forking without the Cisco UCM topology. This topology supports static configuration on CUBE and the replication of media packets to allow recording caller-agent and full-duplex interactions at an IP call recording server.
SIP Recorder Interface

SIP is used as a protocol between CUBE and the MediaSense SIP server. Extensions are made to SIP to carry the recording session information needed for the recording server. This information carried in SIP sessions between the recording client and the recording server is called metadata.

Metadata

Metadata is the information that is passed by the recording client to the recording server in a SIP session. Metadata describes the communication session and its media streams.

Metadata is used by the recording server to:

• Identify participants of the call.
• Associate media streams with the participant information. Each participant can have one or more media streams, such as audio and video.
• Identify the participant change due to transfers during the call.

The recording server uses the metadata information along with other SIP message information, such as dialog ID and time and date header, to derive a unique key. The recording server uses this key to store media streams and associate the participant information with the media streams.
How to Configure Network-Based Recording

Configuring Network-Based Recording (with Media Profile Recorder)

SUMMARY STEPS

1. enable
2. configure terminal
3. media profile recorder profile-tag
4. (Optional) media-type audio
5. media-recording dial-peer-tag [dial-peer-tag2...dial-peer-tag5]
6. exit
7. media class tag
8. recorder profile tag
9. exit
10. dial-peer voice dummy-recorder-dial-peer-tag voip
11. media-class tag
12. destination-pattern [+] string [T]
13. session protocol sipv2
14. session target ipv4:[recording-server-destination-address | recording-server-dns]
15. session transport tcp
16. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> media profile recorder profile-tag</td>
<td>Configures the media profile recorder and enters media profile configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device(config)# media profile recorder 100</td>
<td></td>
</tr>
</tbody>
</table>
## Network-Based Recording

### Configuring Network-Based Recording (with Media Profile Recorder)

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 4</strong></td>
<td>Configures recording of audio only in a call with both audio and video. If this configuration is not done, both audio and video are recorded.</td>
</tr>
</tbody>
</table>

**Example:**

Device(cfg-mediaprofile)# media-type audio

<table>
<thead>
<tr>
<th><strong>Step 5</strong></th>
<th>Configures the dial-peers that need to be configured.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>media-recording dial-peer-tag</strong> [dial-peer-tag2...dial-peer-tag5]</td>
<td>Note You can specify a maximum of five dial-peer tags.</td>
</tr>
</tbody>
</table>

**Example:**

Device(cfg-mediaprofile)# media-recording 8000 8001 8002

<table>
<thead>
<tr>
<th><strong>Step 6</strong></th>
<th>Exits media profile configuration mode.</th>
</tr>
</thead>
</table>

**Example:**

Device(cfg-mediaprofile)# exit

<table>
<thead>
<tr>
<th><strong>Step 7</strong></th>
<th>Configures a media class and enters media class configuration mode.</th>
</tr>
</thead>
</table>

**Example:**

Device(config)# media class 100

<table>
<thead>
<tr>
<th><strong>Step 8</strong></th>
<th>Configures the media profile recorder.</th>
</tr>
</thead>
</table>

**Example:**

Device(cfg-mediaclass)# recorder profile 100

<table>
<thead>
<tr>
<th><strong>Step 9</strong></th>
<th>Exits media class configuration mode.</th>
</tr>
</thead>
</table>

**Example:**

Device(cfg-mediaclass)# exit

<table>
<thead>
<tr>
<th><strong>Step 10</strong></th>
<th>Configures a recorder dial peer and enters dial peer voice configuration mode.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>dial-peer voice dummy-recorder-dial-peer-tag voip</strong></td>
<td></td>
</tr>
</tbody>
</table>

**Example:**

Device(config)# dial-peer voice 8000 voip

<table>
<thead>
<tr>
<th><strong>Step 11</strong></th>
<th>Configures media class on a dial peer.</th>
</tr>
</thead>
</table>

**Example:**

Device(config-dial-peer)# media-class 100

<table>
<thead>
<tr>
<th><strong>Step 12</strong></th>
<th>Specifies either the prefix or the full E.164 telephone number (depending on your dial plan) to be used for a dial peer.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>destination-pattern [+] string [T]</strong></td>
<td></td>
</tr>
</tbody>
</table>

**Example:**

Device(config-dial-peer)# destination-pattern 595959
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 13</strong></td>
<td>session protocol sipv2</td>
</tr>
<tr>
<td>Example: Device(config-dial-peer)# session protocol sipv2</td>
<td>Configures the VoIP dial peer to use Session Initiation Protocol (SIP).</td>
</tr>
<tr>
<td><strong>Step 14</strong></td>
<td>session target ipv4: [recording-server-destination-address</td>
</tr>
<tr>
<td>Example: Device(config-dial-peer)# session target ipv4: 10.42.29.7</td>
<td>Specifies a network-specific address for a dial peer. Keyword and argument are as follows: • ipv4: destination address --IP address of the dial peer, in this format: xxx.xxx.xxx.xxx</td>
</tr>
<tr>
<td><strong>Step 15</strong></td>
<td>session transport tcp</td>
</tr>
<tr>
<td>Example: Device(config-dial-peer)# session transport tcp</td>
<td>Configures a VoIP dial peer to use Transmission Control Protocol (TCP).</td>
</tr>
<tr>
<td><strong>Step 16</strong></td>
<td>end</td>
</tr>
<tr>
<td>Example: Device(config-dial-peer)# end</td>
<td>Returns to privileged EXEC mode.</td>
</tr>
</tbody>
</table>
Configuring Network-Based Recording (without Media Profile Recorder)

SUMMARY STEPS

1. enable
2. configure terminal
3. media class tag
4. recorder parameter
5. (Optional) media-type audio
6. media-recording dial-peer-tag
7. exit
8. exit
9. dial-peer voice dummy-recorder-dial-peer-tag voip
10. media-class tag
11. destination-pattern [+ string [T]
12. session protocol sipv2
13. session target ipv4:[recording-server-destination-address | recording-server-dns]
14. session transport tcp
15. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> media class tag</td>
<td>Configures the media class and enters media class</td>
</tr>
<tr>
<td>Example:</td>
<td>configuration mode.</td>
</tr>
<tr>
<td>Device(config)# media class 100</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> recorder parameter</td>
<td>Enters media class recorder parameter configuration mode to enable you to configure recorder-specific parameters.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device(cfg-mediaclass)# recorder parameter</td>
<td></td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
</tr>
<tr>
<td>------</td>
<td>------------------</td>
</tr>
<tr>
<td>5</td>
<td>media-type audio</td>
</tr>
<tr>
<td>6</td>
<td>media-recording dial-peer-tag</td>
</tr>
<tr>
<td>7</td>
<td>exit</td>
</tr>
<tr>
<td>8</td>
<td>exit</td>
</tr>
<tr>
<td>9</td>
<td>dial-peer voice dummy-recorder-dial-peer-tag voip</td>
</tr>
<tr>
<td>10</td>
<td>media-class tag</td>
</tr>
<tr>
<td>11</td>
<td>destination-pattern [+ string [T]]</td>
</tr>
<tr>
<td>12</td>
<td>session protocol sipv2</td>
</tr>
<tr>
<td>13</td>
<td>session target ipv4:[recording-server-destination-address</td>
</tr>
</tbody>
</table>

---

**Example:**
- Device(config-dial-peer)# media-type audio
- Device(cfg-mediaprofile)# media-type audio
- Device(cfg-mediaclass-recorder)# media-recording dial-peer-tag
- Device(cfg-mediaclass-recorder)# exit
- Device(cfg-mediaclass)# exit
- Device(config)# dial-peer voice 8000 voip
- Device(config-dial-peer)# media-class 100
- Device(config-dial-peer)# destination-pattern 595959
- Device(config-dial-peer)# session protocol sipv2
- Device(config-dial-peer)# session target
### Purpose

#### Command or Action

**Example:**

Device(config-dial-peer)# session target ipv4:10.42.29.7

- **ipv4**: destination address --IP address of the dial peer, in this format: xxx.xxx.xxx.xxx

#### Step 14

**session transport tcp**

**Example:**

Device(config-dial-peer)# session transport tcp

Configures a VoIP dial peer to use Transmission Control Protocol (TCP).

#### Step 15

**end**

**Example:**

Device(config-dial-peer)# end

Returns to privileged EXEC mode.

---

### Verifying the Network-Based Recording Using CUBE

Perform this task to verify the configuration of the Network-Based Recording Using CUBE. The `show` and `debug` commands can be entered in any order.

#### SUMMARY STEPS

1. enable
2. show voip rtp connections
3. show voip recmsp session
4. show voip recmsp session detail call-id call-id
5. show voip rtp forking
6. show call active voice compact
7. show call active video compact
8. show sip-ua calls
9. show call active video brief
10. debug ccsip messages (for audio calls)
11. debug ccsip messages (for video calls)
12. debug ccsip messages (for audio-only recording in a call with both audio and video)
13. Enter one of the following:
   - `debug ccsip all`
   - `debug voip recmsp all`
   - `debug voip ccap all`
   - `debug voip fpi all` (for ASR devices only)
DETAILED STEPS

Step 1  enable
Enables privileged EXEC mode.

Example:
Device> enable

Step 2  show voip rtp connections
 Displays Real-Time Transport Protocol (RTP) connections. Two extra connections are displayed for forked legs.

Example:
Device# show voip rtp connections

VoIP RTP Port Usage Information:
Max Ports Available: 8091, Ports Reserved: 101, Ports in Use: 8
Port range not configured, Min: 16384, Max: 32767

<table>
<thead>
<tr>
<th>Media-Address Range</th>
<th>Ports Available</th>
<th>Ports Reserved</th>
<th>Ports In-use</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>8091</td>
<td>101</td>
<td>8</td>
</tr>
</tbody>
</table>

VoIP RTP active connections:

<table>
<thead>
<tr>
<th>No. CallId</th>
<th>dstCallId</th>
<th>LocalRTP</th>
<th>RmtRTP</th>
<th>LocalIP</th>
<th>RemoteIP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>16384</td>
<td>20918</td>
<td>10.104.45.191</td>
<td>10.104.8.94</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>16386</td>
<td>17412</td>
<td>10.104.45.191</td>
<td>10.104.8.98</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>16388</td>
<td>29652</td>
<td>10.104.45.191</td>
<td>10.104.8.98</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>16390</td>
<td>20036</td>
<td>10.104.45.191</td>
<td>10.104.8.94</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>16392</td>
<td>58368</td>
<td>10.104.45.191</td>
<td>10.104.105.232</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>16394</td>
<td>53828</td>
<td>10.104.45.191</td>
<td>10.104.105.232</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>16396</td>
<td>39318</td>
<td>10.104.45.191</td>
<td>10.104.105.232</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>16398</td>
<td>41114</td>
<td>10.104.45.191</td>
<td>10.104.105.232</td>
</tr>
</tbody>
</table>

Found 8 active RTP connections

Step 3  show voip recmsp session
Displays active recording Media Service Provider (MSP) session information internal to CUBE.

Example:
Device# show voip recmsp session

RECMSP active sessions:
MSP Call-ID | AnchorLeg Call-ID | ForkedLeg Call-ID
-------------|-------------------|-------------------
143          | 141               | 145               
Found 1 active sessions

Step 4  show voip recmsp session detail call-id call-id
Displays detailed information about the recording MSP Call ID.
Example:

```
Device# show voip recmsp session detail call-id 145
RECMSP active sessions:
Detailed Information
--------------------------
Recording MSP Leg Details:
Call ID: 143
GUID : 7C5946D38ECD

Anchor Leg Details:
Call ID: 141
Forking Stream type: voice-nearend
Participant: 708090

Non-anchor Leg Details:
Call ID: 140
Forking Stream type: voice-farend
Participant: 10000

Forked Leg Details:
Call ID: 145
Near End Stream CallID 145 Stream State ACTIVE
Far End stream CallID 146 Stream State ACTIVE
Found 1 active sessions
Device# show voip recmsp session detail call-id 5
```

```
RECMSP active sessions:
Detailed Information
--------------------------
Recording MSP Leg Details:
Call ID: 5
GUID : 1E01B6000000

Anchor Leg Details:
Call ID: 1
Forking Stream type: voice-nearend
Forking Stream type: video-nearend
Participant: 1777

Non-anchor Leg Details:
Call ID: 2
Forking Stream type: voice-farend
Forking Stream type: video-farend
Participant: 1888

Forked Leg Details:
Call ID: 6
Voice Near End Stream CallID 6 Stream State ACTIVE
Voice Far End stream CallID 7 Stream State ACTIVE
Video Near End stream CallID 8 Stream State ACTIVE
Video Far End stream CallID 9 Stream State ACTIVE
Found 1 active sessions
```

<table>
<thead>
<tr>
<th>Output Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stream State</td>
<td>Displays the state of the call. This can be ACTIVE or HOLD.</td>
</tr>
</tbody>
</table>
### Output Field | Description
--- | ---
Msp Call-Id | Displays an internal Media service provider call ID and forking related statistics for an active forked call.
Anchor Leg Call-id | Displays an internal anchor leg ID, which is the dial peer where forking enabled. The output displays the participant number and stream type. Stream type voice-near end indicates the called party side.
Non-Anchor Call-id | Displays an internal non-anchor leg ID, which is the dial peer where forking is not enabled. The output displays the participant number and stream type. Stream type voice-near end indicates the called party side.
Forked Call-id | This forking leg call-id will show near-end and far-end stream call-id details with state of the Stream .
| Displays an internal foked leg ID. The output displays near-end and far-end details of a stream.

### Step 5
**show voip rtp forking**
Displays RTP media-forking connections.

**Example:**

```
Device# show voip rtp forking
VoIP RTP active forks :
 Fork 1
    stream type voice-only (0): count 0
    stream type voice+dtmf (1): count 0
    stream type dtmf-only (2): count 0
    stream type voice-nearend (3): count 1
    remote ip 10.42.29.7, remote port 38526, local port 18648
       codec g711ulaw, logical ssrc 0x53
       packets sent 29687, packets received 0
    stream type voice+dtmf-nearend (4): count 0
    stream type voice-farend (5): count 1
    remote ip 10.42.29.7, remote port 50482, local port 17780
       codec g711ulaw, logical ssrc 0x55
       packets sent 29686, packets received 0
    stream type voice+dtmf-farend (6): count 0
    stream type video (7): count
```

| Output Field | Description |
--- | ---|
remote ip 10.42.29.7, remote port 38526, local port 18648 | Recording server IP, recording server port, and local CUBE device port where data for stream 1 was first sent from. |
remote ip 10.42.29.7, remote port 50482, local port 17780 | Recording server IP, recording server port, and local CUBE device port where data for stream 2 was first sent from. |
packets sent 29686 | Number of packets sent to the recorder |
### Step 6 show call active voice compact
Displays a compact version of voice calls in progress. An additional call leg is displayed for media forking.

**Example:**
```
Device# show call active voice compact
<callID> A/O FAX T<sec> Codec type Peer Address IP R<ip>:<udp>
Total call-legs: 3
  140 ANS T644 g711ulaw VOIP P10000 10.42.30.32:18638
  141 ORG T644 g711ulaw VOIP P708090 10.42.30.189:26184
  145 ORG T643 g711ulaw VOIP P595959 10.42.29.7:38526
```

### Step 7 show call active video compact
Displays a compact version of video calls in progress.

**Example:**
```
Device# show call active video compact
<callID> A/O FAX T<sec> Codec type Peer Address IP R<ip>:<udp>
Total call-legs: 3
  1 ANS T14 H264 VOIP-VIDEO P1777 10.104.8.94:20036
  2 ORG T14 H264 VOIP-VIDEO P1888 10.104.8.98:29652
  6 ORG T13 H264 VOIP-VIDEO P1234 10.104.105.232:39318
```

### Step 8 show sip-ua calls
Displays active user agent client (UAC) and user agent server (UAS) information on SIP calls.

**Example:**
```
Device# show sip-ua calls
Total SIP call legs:3, User Agent Client:2, User Agent Server:1
SIP UAC CALL INFO
Call 1
SIP Call ID : 99EA5118-506211E0-80C6E01B-4C27AA62@10.42.30.10
  State of the call : STATE_ACTIVE (7)
  Substate of the call : SUBSTATE_NONE (0)
  Calling Number : 10000
  Called Number : 708090
  Bit Flags : 0xC04018 0x10000100 0x80
  CC Call ID : 141
  Source IP Address (Sig ) : 10.42.30.10
  Destn SIP Req Addr:Port : [10.42.30.5]:5060
  Destn SIP Resp Addr:Port : [10.42.30.5]:5060
  Destination Name : 10.42.30.5
  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 : 141
  Stream Type : voice+dtmf (1)
  Stream Media Addr Type : 1
  Negotiated Codec : g711ulaw (160 bytes)
  Codec Payload Type : 0
  Negotiated Dtmf-relay : rtp-nte
```
Destination Name: 10.42.30.32
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: 140
Stream Type: voice+dtmf (0)
Stream Media Addr Type: 1
Negotiated Codec: g711ulaw (160 bytes)
Codec Payload Type: 0
Negotiated Dtmf-relay: rtp-nte
Dtmf-relay Payload Type: 101
QoS ID: -1
Local QoS Strength: BestEffort
Negotiated QoS Strength: BestEffort
Negotiated QoS Direction: None
Local QoS Status: None
Media Source IP Addr:Port: [10.42.30.10]:18792
Media Dest IP Addr:Port: [10.42.30.32]:18638

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

Step 9
show call active video brief
Displays a truncated version of video calls in progress.

Example:
Device# show call active video brief

Telephony call-legs: 0
SIP call-legs: 3
H323 call-legs: 0
Call agent controlled call-legs: 0
SCCP call-legs: 0
Multicast call-legs: 0
Total call-legs: 3

0 : 1 87424920ms.1 (*12:23:53.573 IST Wed Jul 17 2013) +1050 pid:1 Answer 1777 active
dur 00:00:46 tx:5250/1857831 rx:5293/1930598 dscp:0 media:0 audio tos:0xB8 video tos:0x88
IP 10.104.8.94:20036 SRTP: off rtt:0ms pl:0/0ms lost:0/0/0 delay:0/0/0ms H264 TextRelay: off
Transcoded: No

0 : 2 87424930ms.1 (*12:23:53.583 IST Wed Jul 17 2013) +1040 pid:2 Originate 1888 active
dur 00:00:46 tx:5250/1857831 rx:5293/1930598 dscp:0 media:0 audio tos:0xB8 video tos:0x88
IP 10.104.8.98:29652 SRTP: off rtt:0ms pl:0/0ms lost:0/0/0 delay:0/0/0ms H264 TextRelay: off
Transcoded: No

0 : 6 87425990ms.1 (*12:23:54.643 IST Wed Jul 17 2013) +680 pid:1234 Originate 1234 active
dur 00:00:46 tx:10398/3732871 rx:0/0 dscp:0 media:0 audio tos:0xB8 video tos:0x0
IP 10.104.105.232:39318 SRTP: off rtt:0ms pl:0/0ms lost:0/0/0 delay:0/0/0ms H264 TextRelay: off
Transcoded: No

Step 10
debug csisip messages (for audio calls)
Sent:
INVITE sip:22222@10.42.30.10 SIP/2.0
Via: SIP/2.0/TCP 10.42.30.10:5060;branch=z9hG4bKB622CF
X-Cisco-Recording-Participant: sip:708090@10.42.30.5;media-index="0"
X-Cisco-Recording-Participant: sip:100000@10.42.30.32;media-index="1"
From: <sip:10.42.30.10>;tag=5096700-1E1A
To: <sip:599598@10.42.29.7>
Date: Fri, 18 Mar 2011 07:10:50 GMT
Call-ID: B6C6C389-506411E0-80EAE01B-4C27A628@10.42.30.10
Supported: 100rel, timer, resource-priority, replaces, sdp-anat
Min-SE: 1800
Cisco-Guid: 1334370502-134899760-8096699092-3395863316
Verifying the Network-Based Recording Using CUBE

User-Agent: Cisco-SIPGateway/IOS-15.2(0.0.2)PIA16
Allow: INVITE, OPTIONS, BYE, CANCEL, ACK, PRACK, UPDATE, REFER, SUBSCRIBE, NOTIFY, INFO, REGISTER
CSeq: 101 INVITE
Max-Forwards: 70
Timestamp: 1300431710
Contact: <sip:10.42.30.10:5060;transport=tcp>
Expires: 180
Allow-Events: telephone-event
Content-Type: application/sdp
Content-Disposition: session;handling=required
Content-Length: 449
v=0
o=CiscoSystemsSIP-GW-UserAgent 3021 3526 IN IP4 10.42.30.10
s=SIP Call
c=IN IP4 10.42.30.10
t=0 0
m=audio 24544 RTP/AVP 0 101 19
m=audio 31166 RTP/AVP 0 101 19
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=rtpmap:19 CN/8000
a=ptime:20
a=sendonly
Received:
SIP/2.0 200 Ok
Via: SIP/2.0/TCP 10.104.46.198:5060;branch=z9hG4bK13262B
To: <sip:23232323@10.104.46.201>;tag=ds457251f
From: <sip:10.104.46.198>;tag=110B66-1CBC
Call-ID: 7142FB-9A5011E0-801EF71A-59B4D258@10.104.46.198
CSeq: 101 INVITE
Content-Type: application/sdp
Content-Disposition: session;handling=required
Content-Length: 0
v=0
o=CiscoORA 2187 1 IN IP4 10.104.46.201
s=SIP Call
c=IN IP4 10.104.46.201
t=0 0
m=audio 54100 RTP/AVP 0
m=audio 39674 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=recvonly
m=audio 39674 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=recvonly
Sent:
ACK sip:23232323@10.104.46.201:5060;transport=tcp SIP/2.0
Via: SIP/2.0/TCP 10.104.46.198:5060;branch=z9hG4bK141B87
From: <sip:10.104.46.198>;tag=110B66-1CBC
To: <sip:23232323@10.104.46.201>;tag=ds457251f
Date: Mon, 20 Jun 2011 08:42:01 GMT
Call-ID: 7142FB-9A5011E0-801EF71A-59B4D258@10.104.46.198
CSeq: 101 ACK
Allow-Events: telephone-event
Content-Length: 0
### Output Field | Description
--- | ---
INVITE sip:22222@10.42.29.7:5060 SIP/2.0 | 22222 is the destination pattern or the number of recording server and is configured under the recorder dial peer.
X-Cisco-Recording-Participant: sip:708090@10.42.30.5;media-index="0" | Cisco proprietary header with originating and terminating participant number and IP address used to communicate to the recording server
Cisco-Guid: 1334370502-1348997600-2396699092-3395863316 | GUID is the same for the primary call and forked call.
m=audio 24544 RTP/AVP 0 101 19 | First m-line of participant with payload type and codec information
m=audio 31166 RTP/AVP 0 101 19 | Second m-line of another participant with codec info and payload type.
a=sendonly | CUBE is always in send only mode towards Recording server.
a=recvonly | Recording server is in receive mode only.

### Step 11
**debug ccsip messages** *(for video calls)*

Sent: INVITE sip:575757@9.45.38.39:7686 SIP/2.0

```
  .
Via: SIP/2.0/UDP 9.41.36.41:5060;branch=z9hG4bK2CC2408
X-Cisco-Recording-Participant: sip:1777@10.104.45.207;media-index="0 2"
X-Cisco-Recording-Participant: sip:1888@10.104.45.207;media-index="1 3"
Cisco-Guid: 0884935168-0000065536-0000000401-3475859466
  .
v=0
  .
  .
m=audio 17232 RTP/AVP 0 19
  .
a=sendonly
m=audio 17234 RTP/AVP 0 19
  .
a=sendonly
m=video 17236 RTP/AVP 126
  .
```

---

*Verifying the Network-Based Recording Using CUBE*
Verifying the Network-Based Recording Using CUBE

Output Field | Description |
--- | --- |
Sent: INVITE sip:575757@9.45.38.39:7686 SIP/2.0 | 22222 is the destination pattern or the number of recording server and is configured under the recorder dial peer. |
X-Cisco-Recording-Participant: sip:1777@10.104.45.207;media-index="0 2" X-Cisco-Recording-Participant: sip:1888@10.104.45.207;media-index="1 3" | Cisco proprietary header with originating and terminating participant number and IP address used to communicate to the recording server. |
Cisco-Guid: 0884935168-0000065536-0000000401-3475859466 | GUID is the same for the primary call and forked call. |
m=audio 17232 RTP/AVP 0 19 | First m-line of participant with payload type and audio codec. |
m=audio 17234 RTP/AVP 0 19 | Second m-line of another participant with payload type and audio codec. |
m=video 17236 RTP/AVP 126 | Third m-line of participant with video payload type and codec info. |
m=video 17238 RTP/AVP 126 | Fourth m-line of another participant with video payload type and codec info. |
a=sendonly | CUBE is always in send only mode towards Recording server. |

Receive:
SIP/2.0 200 OK
v=0
m=audio 1592 RTP/AVP 0
a=recvonly
m=audio 1594 RTP/AVP 0
a=recvonly
m=video 1596 RTP/AVP 126
  a=fmtp:97 profile-level-id=420015
  a=recvonly
m=video 1598 RTP/AVP 126
  a=fmtp:126 profile-level-id=420015
  a=recvonly

Sent:
ACK sip:9.45.38.39:7686;transport=UDP SIP/2.0
Via: SIP/2.0/UDP 9.41.36.41:5060;branch=z9hG4bK2CD7
From: <sip:9.41.36.41>;tag=1ECFD128-24DF
To: <sip:575757@9.45.38.39>;tag=16104SIPpTag011
Date: Tue, 19 Mar 2013 11:40:01 GMT
Call-ID: FFFFFFFF91E00FE6-FFFFFFFF8FC011E2-FFFFFFFF824DF469-FFFFFFFFB6661C0689.41.36.41
Max-Forwards: 70
CSeq: 101 ACK
Allow-Events: telephone-event

Content-Length: 0

<table>
<thead>
<tr>
<th>Output Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>m=audio 1592 RTP/AVP 0</td>
<td>First m-line of recording server after it started listening.</td>
</tr>
<tr>
<td>m=audio 1594 RTP/AVP 0</td>
<td>Second m-line of recording server after it started listening.</td>
</tr>
<tr>
<td>m=video 1596 RTP/AVP 126</td>
<td>Third m-line of recording server after it started listening.</td>
</tr>
<tr>
<td>m=video 1598 RTP/AVP 126</td>
<td>Fourth m-line of recording server after it started listening.</td>
</tr>
<tr>
<td>a=recvonly</td>
<td>Recording server in receive only mode.</td>
</tr>
</tbody>
</table>

**Step 12**

**debug ccsip messages** (for audio-only recording in a call with both audio and video)
Displays offer sent to MediaSense having only audio m-lines, when the **media-type audio** command is configured.

Sent:
INVITE sip:54321@9.45.38.39:36212 SIP/2.0
Via: SIP/2.0/UDP 9.41.36.15:5060;branch=z9hG4bK2216B
X-Cisco-Recording-Participant: sip:4321@9.45.38.39;media-index="0"
X-Cisco-Recording-Participant: sip:1111000010@9.45.38.39;media-index="1"
From: <sip:9.41.36.15>;tag=A2C74-5D9
To: <sip:54321@9.45.38.39>....
Content-Type: application/sdp
Content-Disposition: session;handling=required
Content-Length: 337

v=0
o=CiscoSystemsSIP-GW-UserAgent 9849 5909 IN IP4 9.41.36.15
s=SIP Call
c=IN IP4 9.41.36.15
t=0 0
Step 13

Enter one of the following:

- `debug csip all`
- `debug voip recmsp all`
- `debug voip ccapi all`
- `debug voip fpi all` (for ASR devices only)

Displays detailed debug messages.

For Audio:

Media forking initialized:

*Jun 15 10:37:55.404: //103/3E7E90AE8006/SIP/Info/ccsip_trigger_media_forking: MF: Recv Ack & it's Anchor leg. Start MF.*

Media forking started:

*Jun 15 10:37:55.404: //103/3E7E90AE8006/SIP/Info/ccsip_trigger_media_forking: MF: Recv Ack & it's Anchor leg. Start MF.*
*Jun 15 10:37:55.404: //103/3E7E90AE8006/SIP/Info/ccsip_trigger_media_forking: MF: Recv Ack & it's Anchor leg. Start MF.*
*Jun 15 10:37:55.404: //103/3E7E90AE8006/SIP/Info/ccsip_trigger_media_forking: MF: Recv Ack & it's Anchor leg. Start MF.*

Display detailed debug messages.

For Audio:

Media forking initialized:

*Jun 15 10:37:55.404: //103/3E7E90AE8006/SIP/Info/ccsip_trigger_media_forking: MF: Recv Ack & it's Anchor leg. Start MF.*
*Jun 15 10:37:55.404: //103/3E7E90AE8006/SIP/Info/ccsip_trigger_media_forking: MF: Recv Ack & it's Anchor leg. Start MF.*
*Jun 15 10:37:55.404: //103/3E7E90AE8006/SIP/Info/ccsip_trigger_media_forking: MF: Recv Ack & it's Anchor leg. Start MF.*

Media forking started:

*Jun 15 10:37:55.404: //103/3E7E90AE8006/SIP/Info/ccsip_trigger_media_forking: MF: Recv Ack & it's Anchor leg. Start MF.*
*Jun 15 10:37:55.404: //103/3E7E90AE8006/SIP/Info/ccsip_trigger_media_forking: MF: Recv Ack & it's Anchor leg. Start MF.*
*Jun 15 10:37:55.404: //103/3E7E90AE8006/SIP/Info/ccsip_trigger_media_forking: MF: Recv Ack & it's Anchor leg. Start MF.*

Display detailed debug messages.

For Audio:

Media forking initialized:

*Jun 15 10:37:55.404: //103/3E7E90AE8006/SIP/Info/ccsip_trigger_media_forking: MF: Recv Ack & it's Anchor leg. Start MF.*
*Jun 15 10:37:55.404: //103/3E7E90AE8006/SIP/Info/ccsip_trigger_media_forking: MF: Recv Ack & it's Anchor leg. Start MF.*
*Jun 15 10:37:55.404: //103/3E7E90AE8006/SIP/Info/ccsip_trigger_media_forking: MF: Recv Ack & it's Anchor leg. Start MF.*

Media forking started:

*Jun 15 10:37:55.404: //103/3E7E90AE8006/SIP/Info/ccsip_trigger_media_forking: MF: Recv Ack & it's Anchor leg. Start MF.*
*Jun 15 10:37:55.404: //103/3E7E90AE8006/SIP/Info/ccsip_trigger_media_forking: MF: Recv Ack & it's Anchor leg. Start MF.*
*Jun 15 10:37:55.404: //103/3E7E90AE8006/SIP/Info/ccsip_trigger_media_forking: MF: Recv Ack & it's Anchor leg. Start MF.*

Display detailed debug messages.

For Audio:

Media forking initialized:

*Jun 15 10:37:55.404: //103/3E7E90AE8006/SIP/Info/ccsip_trigger_media_forking: MF: Recv Ack & it's Anchor leg. Start MF.*
*Jun 15 10:37:55.404: //103/3E7E90AE8006/SIP/Info/ccsip_trigger_media_forking: MF: Recv Ack & it's Anchor leg. Start MF.*
*Jun 15 10:37:55.404: //103/3E7E90AE8006/SIP/Info/ccsip_trigger_media_forking: MF: Recv Ack & it's Anchor leg. Start MF.*

Media forking started:

*Jun 15 10:37:55.404: //103/3E7E90AE8006/SIP/Info/ccsip_trigger_media_forking: MF: Recv Ack & it's Anchor leg. Start MF.*
*Jun 15 10:37:55.404: //103/3E7E90AE8006/SIP/Info/ccsip_trigger_media_forking: MF: Recv Ack & it's Anchor leg. Start MF.*
*Jun 15 10:37:55.404: //103/3E7E90AE8006/SIP/Info/ccsip_trigger_media_forking: MF: Recv Ack & it's Anchor leg. Start MF.*

Display detailed debug messages.

For Audio:

Media forking initialized:

*Jun 15 10:37:55.404: //103/3E7E90AE8006/SIP/Info/ccsip_trigger_media_forking: MF: Recv Ack & it's Anchor leg. Start MF.*
*Jun 15 10:37:55.404: //103/3E7E90AE8006/SIP/Info/ccsip_trigger_media_forking: MF: Recv Ack & it's Anchor leg. Start MF.*
*Jun 15 10:37:55.404: //103/3E7E90AE8006/SIP/Info/ccsip_trigger_media_forking: MF: Recv Ack & it's Anchor leg. Start MF.*

Media forking started:

*Jun 15 10:37:55.404: //103/3E7E90AE8006/SIP/Info/ccsip_trigger_media_forking: MF: Recv Ack & it's Anchor leg. Start MF.*
*Jun 15 10:37:55.404: //103/3E7E90AE8006/SIP/Info/ccsip_trigger_media_forking: MF: Recv Ack & it's Anchor leg. Start MF.*
*Jun 15 10:37:55.404: //103/3E7E90AE8006/SIP/Info/ccsip_trigger_media_forking: MF: Recv Ack & it's Anchor leg. Start MF.*

Display detailed debug messages.

For Audio:

Media forking initialized:

*Jun 15 10:37:55.404: //103/3E7E90AE8006/SIP/Info/ccsip_trigger_media_forking: MF: Recv Ack & it's Anchor leg. Start MF.*
*Jun 15 10:37:55.404: //103/3E7E90AE8006/SIP/Info/ccsip_trigger_media_forking: MF: Recv Ack & it's Anchor leg. Start MF.*
*Jun 15 10:37:55.404: //103/3E7E90AE8006/SIP/Info/ccsip_trigger_media_forking: MF: Recv Ack & it's Anchor leg. Start MF.*

Media forking started:

*Jun 15 10:37:55.404: //103/3E7E90AE8006/SIP/Info/ccsip_trigger_media_forking: MF: Recv Ack & it's Anchor leg. Start MF.*
*Jun 15 10:37:55.404: //103/3E7E90AE8006/SIP/Info/ccsip_trigger_media_forking: MF: Recv Ack & it's Anchor leg. Start MF.*
*Jun 15 10:37:55.404: //103/3E7E90AE8006/SIP/Info/ccsip_trigger_media_forking: MF: Recv Ack & it's Anchor leg. Start MF.*

Display detailed debug messages.
*Jun 15 10:37:55.404: //103/3E7E90AE8006/SIP/Info/ccsip_ipip_media_forking: MF: Current State = 1, event =30
*Jun 15 10:37:55.404: //103/3E7E90AE8006/SIP/Info/ccsip_ipip_media_forking: MF: State & Event combination is cracked..
*Jun 15 10:37:55.404: //103/3E7E90AE8006/SIP/Function/sipSPIGetMainStream:
*Jun 15 10:37:55.404: //103/3E7E90AE8006/SIP/Function/sipSPIGetMainStream:

Forking header populated:


Media forking setup record session is successful:

*Jun 15 10:37:55.404: //103/3E7E90AE8006/SIP/Info/ccsip_get_recording_participant_header: MF: Sipuser = 9.42.30.34
*Jun 15 10:37:55.404: //103/3E7E90AE8006/SIP/Function/sipSPIGetFirstStream:
*Jun 15 10:37:55.404: //103/3E7E90AE8006/SIP/Function/voip_media_dir_to_cc_media_dir:
*Jun 15 10:37:55.404: //103/3E7E90AE8006/SIP/Info/ccsip_ipip_media_forking_BuildMediaRecSession: MF: direction type = 3
*Jun 15 10:37:55.404: //103/3E7E90AE8006/SIP/Info/ccsip_ipip_media_forking_BuildMediaRecSession: MF: callid 103 set to nearend..
*Jun 15 10:37:55.404: //103/3E7E90AE8006/SIP/Info/ccsip_ipip_media_forking_BuildMediaRecSession: MF: dtmf is inband
*Jun 15 10:37:55.404: //103/3E7E90AE8006/SIP/Info/ccsip_ipip_media_forking_BuildMediaRecSession: MF: First element..
*Jun 15 10:37:55.404: //103/3E7E90AE8006/SIP/Info/ccsip_ipip_media_forking_BuildMediaRecRecParticipant: MF: First element..
*Jun 15 10:37:55.404: //103/3E7E90AE8006/SIP/Info/ccsip_ipip_media_forking_write_to_TDContainer: MF: Data written to TD Container..
*Jun 15 10:37:55.404: //1-xxxxx/Inout/recmsp_api_setup_session: Exit with Success

Media forking forked stream started:

*Jun 15 10:37:55.404: //106/xxxxxxxxxxxx/CCAPI/cc_set_post_tagdata:
*Jun 15 10:37:55.406: //106/000000000000/SIP/Info/ccsip_ipip_media_forking_forked_leg_config: MF: Overwriting the GUID with the value got from MSP.
*Jun 15 10:37:55.406: //106/000000000000/SIP/Info/ccsip_lwf_handle_peer_event:
*Jun 15 10:37:55.406: //106/000000000000/SIP/Info/ccsip_lwf_process_event:
*Jun 15 10:37:55.406: //106/000000000000/SIP/Info/sipSPIUIsValidCb:
*Jun 15 10:37:55.406: //103/3E7E90AE8006/SIP/Info/ccsip_ipip_media_forking_read_from_TDContainer: MF: Data read from TD container
Verifying the Network-Based Recording Using CUBE

For Video:
Media Forking Initialized:

*Mar 19 16:40:01.784 IST: //522/34BF0A000000/SIP/Info/info/36864/ccsip_ipip_media_forking: MF: Current State = 1, event =31 & Event combination is cracked..

Media forking started:


Recording participant for anchor leg:


Adding an audio stream:

*Mar 19 16:40:01.788 IST: //522/34BF0A000000/SIP/Function/sipSPIGetMainStream:

*Mar 19 16:40:01.789 IST: //522/34BF0A000000/SIP/Info/info/32768/ccsip_ipip_media_forking_precondition: MF: Call can be started with current config.

*Mar 19 16:40:01.787 IST: //522/34BF0A000000/SIP/Info/info/32816/ccsip_get_recording_participant_header: MF: Setting data for audio stream..

Video forking:

*Mar 19 16:40:01.789 IST: //522/34BF0A000000/SIP/Function/sipSPIGetVideoStream:

# Additional References for Network-Based Recording

## Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>MediaSense Installation and Administration Guide</td>
<td>Cisco MediaSense Installation and Administration Guide</td>
</tr>
</tbody>
</table>

## Standards and RFCs

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC 3984</td>
<td><em>RTP Payload Format for H.264 Video</em></td>
</tr>
<tr>
<td>RFC 5104</td>
<td><em>Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF)</em></td>
</tr>
<tr>
<td>RFC 5168</td>
<td><em>XML Schema for Media Control</em></td>
</tr>
</tbody>
</table>
CHAPTER 11

Video Recording - Additional Configurations

This module describes the following additional configurations that can be done for Video Recording:

- Request a Full-Intra Frame using RTCP or SIP INFO methods.
- Configure an H.264 Packetization mode.
- Monitor Intra-Frames and Reference Frames

- Feature Information for Video Recording - Additional Configurations, page 109
- Information About Additional Configurations for Video Recording, page 110
- How to Configure Additional Configurations for Video Recording, page 111
- Verifying Additional Configurations for Video Recording, page 114

Feature Information for Video Recording - Additional Configurations

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 . An account on Cisco.com is not required.
### Table 12: Feature Information for Network-Based Recording of Video Calls Using Cisco Unified Border Element

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network-Based Recording of Video Calls Using Cisco Unified Border Element</td>
<td>15.3(3)M Cisco IOS XE Release 3.10S</td>
<td>The Network-Based Recording of Video Calls Using Cisco Unified Border Element feature supports software-based forking and recording of video calls. The following commands were introduced or modified: media profile video, ref-frame-req rtcp, ref-frame-req sip-info, video profile, h264-packetization-mode, monitor-ref-frames.</td>
</tr>
</tbody>
</table>

### Information About Additional Configurations for Video Recording

#### Full Intra-Frame Request

Full Intra-Frame Request is a request sent for an I-frame. An I-frame is an entire key or reference frame that is compressed without considering preceding or succeeding video frames. Succeeding video frames are differences to the original I-frame (what has moved) instead of entire video frame information.

The call between Cisco Unified Border Element and the Cisco MediaSense server is established after the call between the endpoints is established. As a result, the Real-Time Transport Protocol (RTP) channel between the endpoints gets established first and the RTP channel with the recording server gets established later. The impact of this delay is more on video recording because the initial I-frame from the endpoint may not get forked, and frames that follow cannot get decoded. To mitigate the impact of the lost RTP video packets, Cisco Unified Border Element generates Full Intra-Frame Request (FIR) using either Real-Time Transport Control Protocol (RTCP) or SIP INFO, or both, requesting the endpoint to send a fully encoded video frame in the subsequent RTP packet.

The following types of FIR are supported on network-based recording of video calls using Cisco Unified Border Element:

- RTCP FIR (based on RFC 5104).
- SIP INFO FIR (based on RFC 5168).
- Both RTCP FIR and SIP INFO FIR (Cisco Unified Border Element can be configured to send both RTCP FIR and SIP INFO requests at the same time).
How to Configure Additional Configurations for Video Recording

Enabling FIR for Video Calls (Using RTCP of SIP INFO)

Perform this task to enable Full Intra-Frame Request (FIR) during the network-based recording of a video call using Real-Time Transport Control Protocol (RTCP) or using the Session Initiation Protocol (SIP) INFO method.

SUMMARY STEPS

1. enable
2. configure terminal
3. media profile video media-profile-tag
4. Do one of the following:
   • ref-frame-req rtcp retransmit-count retransmit-number
   • ref-frame-req sip-info
5. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example: Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> media profile video media-profile-tag</td>
<td>Configures a video media profile and enters media profile configuration mode.</td>
</tr>
<tr>
<td>Example: Device(config)# media profile video 1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> Do one of the following:</td>
<td>Enables FIR using the RTCP or SIP INFO method.</td>
</tr>
<tr>
<td>• ref-frame-req rtcp retransmit-count retransmit-number</td>
<td></td>
</tr>
<tr>
<td>• ref-frame-req sip-info</td>
<td></td>
</tr>
<tr>
<td>Example: Device(cfg-mediaprofile)# ref-frame-req rtcp retransmit-count 4</td>
<td></td>
</tr>
</tbody>
</table>
### Configuring H.264 Packetization Mode

When a device configured as CUBE is offered more than one H.264 packetization mode on an inbound video call leg, the device offers all received modes to the outbound call leg, allowing dynamic change of mode during a call. However, when a call is forked, the MediaSense recording server is not able to support this dynamic change of the packetization mode. This feature restricts the device and allows it to offer only the configured packetization mode to the outbound call leg when media forking is configured.

#### SUMMARY STEPS

1. enable
2. configure terminal
3. media profile video media-profile-tag
4. h264-packetization-mode packetization mode
5. end

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>Purpose</strong></td>
</tr>
<tr>
<td>enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example: Device&gt; enable</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 2</th>
<th>configure terminal</th>
<th>Enters global configuration mode.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example: Device(config)# configure terminal</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 3</th>
<th>media profile video media-profile-tag</th>
<th>Configures a video media profile and enters media profile configuration mode.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example: Device(config)# media profile video 1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Purpose

Configure the H.264 packetization mode offered by a device on the outbound call leg of a forked call when multiple H.264 packetization modes are present in the offer received by the device on the inbound call leg.

### Example

**Device(cfg-mediaprofile)# h264-packetization-mode 2**

### Step 4

### Command or Action

- **h264-packetization-mode packetization mode**

### Purpose

Configures the H.264 packetization mode offered by a device on the outbound call leg of a forked call when multiple H.264 packetization modes are present in the offer received by the device on the inbound call leg.

### Example

**Device(cfg-mediaprofile)# h264-packetization-mode 2**

### Step 5

### Command or Action

- **end**

### Purpose

Exits media profile configuration mode.

### Example

**Device(cfg-mediaprofile)# end**

---

### Monitoring Reference files or Intra Frames

Perform this task to configure device to perform deep packet inspection (DPI) of RTP packets received from an endpoint and keep track of how many instantaneous decoder refresh (IDR) frames have been received and the timestamp of the IDR.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **media profile video media-profile-tag**
4. **monitor-ref-frames**
5. **end**

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>enable</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>Device&gt; enable</strong></td>
</tr>
<tr>
<td></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><strong>configure terminal</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>Device# configure terminal</strong></td>
</tr>
<tr>
<td></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><strong>media profile video media-profile-tag</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><strong>Device(config)# media profile video l</strong></td>
</tr>
<tr>
<td></td>
<td>Configures a video media profile and enters media profile configuration mode.</td>
</tr>
</tbody>
</table>
### Purpose

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>monitor-ref-frames</td>
<td>Monitors reference frames or intra-frames.</td>
</tr>
<tr>
<td>end</td>
<td>Exits media profile configuration mode.</td>
</tr>
</tbody>
</table>

### Step 4

**Example:**
```
Device(cfg-mediaprofile)# monitor-ref-frames
```

### Step 5

**Example:**
```
Device(cfg-mediaprofile)# end
```

---

## Verifying Additional Configurations for Video Recording

Perform this task to verify the additional configurations of the video recording. The `show` commands can be entered in any order.

### SUMMARY STEPS

1. `enable`
2. `show call active video called-number number | include VideoRtcpIntraFrameRequestCount`
3. `show call active video called-number number | include VideoSipInfoIntraFrameRequestCount`
4. `show call active video | include VideoTimeOfLastReferenceFrame`
5. `show call active video | include VideoReferenceFrameCount`

### DETAILED STEPS

### Step 1

**enable**

Enables privileged EXEC mode.

**Example:**
```
Device> enable
```

### Step 2

**show call active video called-number number | include VideoRtcpIntraFrameRequestCount**

Displays the number of RTCP FIR requests sent on each leg.

**Example:**
```
Device# show call active video called-number 990057 | include VideoRtcpIntraFrameRequestCount

! Main call legs
VideoRtcpIntraFrameRequestCount=1
VideoRtcpIntraFrameRequestCount=1

! CUBE does not generate FIR request on forked leg
VideoRtcpIntraFrameRequestCount=0
```

### Step 3

**show call active video called-number number | include VideoSipInfoIntraFrameRequestCount**

Displays the number of SIP INFO FIR requests sent on each leg.
Example:

```
Device# show call active video called-number 990062 | include VideoSipInfoIntraFrameRequestCount

! Main call legs
VideoSipInfoIntraFrameRequestCount=1
VideoSipInfoIntraFrameRequestCount=0

! CUBE does not generate FIR request on forked leg
```

**Step 4**  
```
show call active video | include VideoTimeOfLastReferenceFrame
```

Displays the timestamp of latest IDR frame.

**Step 5**  
```
show call active video | include VideoReferenceFrameCount
```

Displays the number of IDR frames received on that call leg.
TDos Attack Mitigation

The TDoS Attack Mitigation feature enables Cisco Unified Border Element (Cisco UBE) to not respond to Session Initiation Protocol (SIP) requests from IP addresses that are not listed in a trusted IP address list. Cisco UBE validates only out-of-dialog SIP requests against IP addresses in the trusted IP address list. It does not validate in-dialog SIP requests because such requests usually arrive from trusted entities. The TDoS Attack Mitigation feature is supported both on IPv4 and IPv6 networks.

- Finding Feature Information, page 117
- Information About TDoS Attack Mitigation, page 117
- How to Configure TDoS Attack Mitigation, page 118
- Verifying TDoS Attack Mitigation, page 121
- Configuration Examples for TDoS Attack Mitigation, page 122
- Feature Information for TDoS Attack Mitigation, page 122

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About TDoS Attack Mitigation

The TDoS Attack Mitigation feature prevents Cisco Unified Border Element (Cisco UBE) from responding to Session Initiation Protocol (SIP) requests arriving from untrusted IP addresses, which leads to an improvement in performance. The SIP stack authenticates the source IP address of an incoming SIP request and blocks the response if the source IP address does not match any IP address in the trusted IP address list. To create a trusted IP address list, you may configure a list of IP addresses or use the IP addresses that have been configured using the session target command in dial-peer configuration mode.
Cisco UBE does not respond to REGISTER requests and consumes REGISTER requests if you configure it only for Telephony Denial-of-Service (TDoS) Attack Mitigation and not as a registrar server.

If you configure Cisco UBE as a registrar server for TDoS attack mitigation, it consumes responses for REGISTER requests that do not belong to any application. Cisco UBE does not consume responses to REGISTER requests that belong to a registrar application.

**Note**

A SIP registrar is a server that accepts REGISTER requests and is typically collocated with a proxy or redirect server.

Syslogs are printed on the device console every 60 minutes after Cisco UBE consumes a threshold value of 1000 SIP requests.

### How to Configure TDoS Attack Mitigation

### Configuring a Trusted IP Address List for Toll-Fraud Prevention

**SUMMARY STEPS**

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

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** enable | Enables privileged EXEC mode.  
  Example:  
  `Device> enable`  
  • Enter your password if prompted. |
| **Step 2** configure terminal | Enters global configuration mode.  
  Example:  
  `Device# configure terminal` |
| **Step 3** voice service voip | Enters global VoIP configuration mode.  
  Example:  
  `Device(config)# voice service voip` |
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 4</strong></td>
<td><strong>Purpose</strong></td>
</tr>
<tr>
<td>ip address trusted list</td>
<td>Enters IP address trusted list mode and enables the addition of valid IP addresses.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device(conf-voi-serv)# ip address trusted list</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td><strong>Purpose</strong></td>
</tr>
<tr>
<td>ipv4 ipv4-address [network-mask]</td>
<td>Allows you to add up to 100 IPv4 addresses in the IP address trusted list. Duplicate IP addresses are not allowed.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device(cfg-iptrust-list)# ipv4 192.0.2.1</td>
<td></td>
</tr>
<tr>
<td>255.255.255.0</td>
<td></td>
</tr>
<tr>
<td>• The network-mask argument allows you to define a subnet IP address.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td><strong>Purpose</strong></td>
</tr>
<tr>
<td>ipv6 ipv6-address</td>
<td>Allows you to add IPv6 addresses to the trusted IP address list.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device(cfg-iptrust-list)# ipv6 2001:DB8:0:ABCD::1/48</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td><strong>Purpose</strong></td>
</tr>
<tr>
<td>end</td>
<td>Returns to privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device(cfg-iptrust-list)# end</td>
<td></td>
</tr>
</tbody>
</table>

### Configuring TDoS Attack Mitigation

**SUMMARY STEPS**

1. enable
2. configure terminal
3. voice service voip
4. ip address trusted authenticate
5. allow-connections from-type to to-type
6. sip
7. no registrar server
8. silent-discard untrusted
9. end
10. show sip-ua statistics
11. clear sip-ua statistics

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>enable</td>
</tr>
<tr>
<td></td>
<td>Enables privileged EXEC mode.</td>
</tr>
</tbody>
</table>
### Configuring TDoS Attack Mitigation

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong> Device&gt; enable</td>
<td>Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> voice service voip</td>
<td>Enters voice service configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config)# voice service voip</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> ip address trusted authenticate</td>
<td>Enables IP address authentication on incoming H.323 or Session Initiation Protocol (SIP) trunk calls for toll fraud prevention support.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(conf-voi-serv)# ip address trusted authenticate</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> allow-connections from-type to to-type</td>
<td>Allows connections between specific types of endpoints in a Cisco UBE.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(conf-voi-serv)# allow-connections sip to sip</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> sip</td>
<td>Enters SIP configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(conf-voi-serv)# sip</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> no registrar server</td>
<td>Disables the local SIP registrar.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(conf-serv-sip)# no registrar server</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> silent-discard untrusted</td>
<td>Discards SIP requests from untrusted sources on an incoming SIP trunk.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(conf-serv-sip)# silent-discard untrusted</td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong> end</td>
<td>Returns to privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(conf-serv-sip)# end</td>
<td></td>
</tr>
<tr>
<td><strong>Step 10</strong> show sip-ua statistics</td>
<td>(Optional) Displays response, traffic, and retry SIP statistics.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device# show sip-ua statistics</td>
<td></td>
</tr>
<tr>
<td><strong>Step 11</strong> clear sip-ua statistics</td>
<td>(Optional) Resets the SIP user agent (UA) statistical counters to zero.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device# clear sip-ua statistics</td>
<td></td>
</tr>
</tbody>
</table>
Verifying TDoS Attack Mitigation

Sample output for the show sip-ua statistics command

To display response, traffic, and retry Session Initiation Protocol (SIP) statistics, use the `show sip-ua statistics` command in privileged EXEC mode.

Device# show sip-ua statistics

SIP Response Statistics (Inbound/Outbound)
Informational:
  Trying 0/0, Ringing 0/0, Forwarded 0/0, Queued 0/0,
  SessionProgress 0/0
Success:
  OkInvite 0/0, OkBye 0/0, OkCancel 0/0, OkOptions 0/0,
  OkPrack 0/0, OkRegister 0/0
  OkSubscribe 0/0, OkNotify 0/0, OkPublish 0/0
  OkInfo 0/0, OkUpdate 0/0,
  202Accepted 0/0, OkOptions 0/0
Redirection (Inbound only except for MovedTemp(Inbound/Outbound)):
  MultipleChoice 0, MovedPermanently 0,
  MovedTemporarily 0/0, UseProxy 0,
  AlternateService 0
Client Error:
  BadRequest 0/0, Unauthorized 0/0,
  PaymentRequired 0/0, Forbidden 0/0,
  NotFound 0/0, MethodNotAllowed 0/0,
  NotAcceptable 0/0, ProxyAuthReqd 0/0,
  ReqTimeout 0/0, Conflict 0/0, Gone 0/0,
  ConditionalRequestFailed 0/0,
  ReqEntityTooLarge 0/0, ReqURITooLarge 0/0,
  UnsupportedMediaType 0/0, UnsupportedURIScheme 0/0,
  BadExtension 0/0, IntervalTooBrief 0/0,
  TempNotAvailable 0/0, CallLegNonExistent 0/0,
  LoopDetected 0/0, TooManyHops 0/0,
  AddrIncomplete 0/0, Ambiguous 0/0,
  BusyHere 0/0, RequestCancel 0/0,
  NotAcceptableMedia 0/0, BadEvent 0/0,
  SETooSmall 0/0, RequestPending 0/0,
  UnsupportedResourcePriority 0/0,
  Total untrusted Request Consumed 1500, //This counter increments (+1) on reception of an untrusted SIP request.//
  Untrusted Request Consumed in last lap 300, //This counter is updated after every 60 minutes.//
  Last Threshold for Untrusted Request Consumed 1000 //This counter activates when the router boots up. Counter value is the number of untrusted requests that are consumed (after crossing 1000 SIP requests) in each interval of 60 minutes after the router boots up.//
Server Error:
  InternalError 0/0, NotImplemented 0/0,
  BadGateway 0/0, ServiceUnavailable 0/0,
  GatewayTimeout 0/0, BadSipVer 0/0,
  PreCondFailure 0/0
Global Failure:
  BusyEverywhere 0/0, Decline 0/0,
  NotExistAnywhere 0/0, NotAcceptable 0/0
Miscellaneous counters:
  RedirectRspMappedToClientErr 0

SIP Total Traffic Statistics (Inbound/Outbound)
  Invite 0/0, Ack 0/0, Bye 0/0,
  Cancel 0/0, Options 0/0,
Configuration Examples for TDoS Attack Mitigation

Example: Trusted IP Address List Configuration

The following example shows how to configure a Trusted IP Address list.

Device> enable
Device# configure terminal
Device(config)# voice service voip
Device(conf-voi-serv)# ip address trusted list
Device(cfg-iptrust-list)# ipv4 192.0.2.1
Device(cfg-iptrust-list)# ipv6 2001:DB8:0:ABCD::1/48

Example: TDoS Attack Mitigation Configuration

The following example shows how to configure TDoS Attack Mitigation.

Device> enable
Device# configure terminal
Device(config)# voice service voip
Device(conf-voi-serv)# ip address trusted authenticate
Device(conf-voi-serv)# allow-connections sip to sip
Device(conf-voi-serv)# sip
Device(conf-serv-sip)# no registrar server
Device(conf-serv-sip)# silent-discard untrusted

Feature Information for TDoS Attack Mitigation

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 . An account on Cisco.com is not required.
### Table 13: Feature Information for TDoS Mitigation

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Release</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>TDoS Attack Mitigation</td>
<td>15.3(3)M</td>
<td>The TDoS Attack Mitigation feature enables Cisco UBE to not respond to Session Initiation Protocol (SIP) requests from IP addresses that are not listed in a trusted IP address list.</td>
</tr>
<tr>
<td>TDoS Attack Mitigation</td>
<td>Cisco IOS XE Release 3.10S</td>
<td>The TDoS Attack Mitigation feature enables Cisco UBE to not respond to Session Initiation Protocol (SIP) requests from IP addresses that are not listed in a trusted IP address list.</td>
</tr>
</tbody>
</table>
CHAPTER 13

Cisco Unified Communications Gateway Services--Extended Media Forking

The Cisco Unified Communications (UC) Services API provides a unified web service interface for the different services in IOS gateway thereby facilitating rapid service development at application servers and managed application service providers.

This chapter explains the Extended Media Forking (XMF) provider that allows applications to monitor calls and trigger media forking on Real-time Transport Protocol (RTP) and Secure RTP calls.

- Feature Information for Cisco Unified Communications Gateway Services—Extended Media Forking, page 125
- Restrictions for Unified Communications Gateway Services—Extended Media Forking, page 126
- Information About Cisco Unified Communications Gateway Services, page 126
- How to Configure UC Gateway Services, page 133
- Configuration Examples for UC Gateway Services, page 140

Feature Information for Cisco Unified Communications Gateway Services—Extended Media Forking

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 . An account on Cisco.com is not required.

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco Unified Communications Gateway Services</td>
<td>Cisco IOS 15.3(3)M, Cisco IOS XE 3.10S</td>
<td>The Cisco Unified Communications (UC) Services API provides a unified web service interface for the different services in IOS gateway thereby</td>
</tr>
</tbody>
</table>
## Restrictions for Unified Communications Gateway Services—Extended Media Forking

- Media renegotiation is not supported.
- Media mixing on forked media streams is not supported.
- recordTone insertion is not supported with SRTP calls.
- mediaForkingReason tag is only to notify midcall stream events; notification for events such as codec change is not supported.
- Only voice media stream is supported.
- Supplementary services are not supported.
- High Availability is not supported.

## Information About Cisco Unified Communications Gateway Services

### Extended Media Forking (XMF) Provider and XMF Connection

The XMF provider allows applications to monitor calls and trigger media forking on the calls and has the capability to service up to 32 applications. The XMF provider can invoke a call-based or a connection-based media forking using the Unified Communications (UC) API. After the media forking is invoked, it can preserve the media forking initiated by the web application if the WAN connection to the application is lost. The XMF provider also provides the recording tone to the parties involved in the call.

The XMF connection describes the relationship between an XMF call and the endpoint (or trunk) involved in the call. A connection abstraction maintained in the gateway has the following connection states:

- IDLE: This state is the initial state for all new connections. Such connections are not actively part of a telephone call, yet their references to the Call and Address objects are valid. Connections typically do
not stay in the IDLE state for long and quickly transition to other states. The application may choose to be notified at this state using the event filters and if done, call/connection at the gateway provider will use the NotifyXmfConnectionData(CREATED) message to notify the application listener that a new connection is created.

- ADDRESS_COLLECT: In this state the initial information package is collected from the originating party and is examined according to the “dialing plan” to determine the end of collection of addressing information. In this state, the call in the gateway collects digits from the endpoint. No notification is provided.

- CALL_DELIVERY: On the originating side, this state involves selecting of the route as well as sending an indication of the desire to set up a call to the specified called party. On the terminating side, this state involves checking the busy/idle status of the terminating access and also informing the terminating message of an incoming call. The application may choose to be notified at this state using the event filters and if done, the call or connection at the gateway provider will use the NotifyXmfConnectionData(CALL_DELIVERY) message to notify the application listener.

- ALERTING: This state implies that the Address is being notified of an incoming call. The application may choose to be notified at this state using the event filters and if done, the call or connection at the gateway provider will use the NotifyXmfConnectionData(ALERTING) message to notify the application listener.

- CONNECTED: This state implies that a connection and its Address is actively part of a telephone call. In common terms, two parties talking to one another are represented by two connections in the CONNECTED state. The application may choose to be notified at this state using the event filters and if done, the call or connection at the gateway provider will use the NotifyXmfConnectionData(CONNECTED) message to notify the application listener.

- DISCONNECTED: This state implies it is no longer part of the telephone call. A Connection in this state is interpreted as once previously belonging to this telephone call. The application may choose to be notified at this state using the event filters and if done, the call or connection at the gateway provider will use the NotifyXmfConnectionData(DISCONNECTED) message to notify the application listener.

XMF Call-Based Media Forking

In call-based media forking of the gateway, the stream from the calling party is termed as near-end stream and the stream from the called party is termed as far-end stream. The XMF provider actively handles single media forking request per session. Any new media forking request from the external application will override or stop the current forking instance and would start a new forking instance (to the appropriate target IP address or ports). After the media forking request is accepted, the XMF provider returns a response message and starts to fork media streams of a connection to the target forked streams. A NotifyXmfCallData message will be notified to the application for the updated media forking status, that is, FORK-FAILED, FORK_STARTED, or FORK_DONE.

XMF Connection-Based Media Forking

In connection-based media forking of the gateway, the incoming stream to the connection is termed as near-end stream and the outgoing stream of the connection is termed as far-end stream. The XMF provider actively handles single media forking request per session. Any new media forking request from the external application will override or stop the current forking instance and would start a new forking instance (to the appropriate target IP address or ports). After the media forking request is accepted, the XMF provider returns a response message and starts to fork media streams of the connection to the target forked streams. A NotifyXmfCallData message will be notified to the application for the updated media forking status, that is, FORK-FAILED, FORK_STARTED, or FORK_DONE.
target IP address or ports). After the media forking request is accepted, the XMF provider returns a response message and starts to fork media streams of a connection to the target forked streams.

**Figure 2: XMF Connection-Based Media Forking**

![Diagram of XMF Connection-Based Media Forking]

A NotifyXmfConnectionData message will be notified to the application for the updated media forking status:

- **FORK_FAILED**—Media forking is setup failure. No forked RTP connections can be established to target RTP addresses.
- **FORK_STARTED**—Media forking is set up successfully. Both Tx (transmit) and Rx (receive) forked RTP connections are established and connected to target (farEnd and nearEnd) RTP addresses.
- **FORK_DONE**—Media forking is completed. Both Tx and Rx forked RTP connections are released.

**Cisco UC Gateway Services Media Forking API with Survivability TCL**

Cisco Unified Border Element (CUBE) supports Survivability TCL Script to co-exist with Cisco Unified Communication (UC) Services API.

Cisco UC Services API XMF interface supports media forking for all the calls controlled by survivability TCL script including the survivability re-attempted calls. Thus, all the calls controlled by survivability TCL script can be recorded when requested by Cisco UC Services XMF API.

Cisco Unified Communications Manager controlled Gateway recording utilizes XMF to trigger media forking on CUBE or SIP based PSTN gateways in the supported call flows.

**Note**

Media forking is allowed only for survivability TCL script supported by Cisco Unified Customer Voice Portal (CVP). CVP survivability TCL script is not supported in High Availability mode.
The following call scenarios are supported:

- Basic comprehensive call
- Calls with Refer Consume
- Calls with Mid-call failure
- Calls with alternative route with initial call failure

There are no configuration changes required for enabling CVP survivability TCL support with Cisco UC Gateway Services API.

**Media Forking for SRTP Calls**

- SRTP forking is supported in XCC and XMF application service providers and the supported APIs are RequestCallMediaForking, RequestCallMediaSetAttributes, and RequestConnectionMediaForking.
- SRTP forking is supported for SRTP-to-SRTP, SRTP-to-RTP, and RTP-to-SRTP calls.
  - For SRTP-to-SRTP calls, media forking on either leg would result in SRTP streams being forked.
  - For SRTP fallback calls, after the initial offer, CUBE will fall back to RTP. Media forking either call legs would result in RTP streams being forked.
  - For SRTP-to-RTP interworking calls, a digital signal processor (DSP) is required and involves transcoding. In this case, one leg would be SRTP and the other leg RTP.
- SRTP Crypto keys are notified over the API.
- Supports automatic stopping of media forking when stream changes from SRTP or to SRTP.
  - The optional mediaForkingReason tag in XMF or XCC Notify messages indicates that the forking has been stopped internally.
  - mediaForkingReason tag is only present when the connection changes state, such as mid-call re-INVITE. SRTP stream can change to RTP or SRTP stream can change keys mid-call.
  - mediaForkingReason tag is always accompanied by FORK_DONE.

**Crypto Tag**

For SRTP forking, the optional Crypto tag in NotifyXmfConnectionData or NotifyXmfCallData message indicates the context of an actively forked SRTP connection.

- The Crypto tag is only present in the notification message where FORK_STARTED tag is present.

The optional Crypto tag specifies the following:

- The Crypto suite used for encryption and authentication algorithm.
- The base64 encoded mastery key and salt used for encryption.
Crypto suite can be one of the two suites supported in IOS:

- AES_CM_128_HMAC_SHA1_32
- AES_CM_128_HMAC_SHA1_80

Example of SDP Data sent in an SRTP Call

<table>
<thead>
<tr>
<th>Original SIP SDP Crypto Offer</th>
<th>SIP SDP Crypto Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>v=0</td>
<td>v=0</td>
</tr>
<tr>
<td>o=CiscoSystemsSIP-GW-UserAgent 7826 3751 IN IP4 172.18.193.98</td>
<td>o=CiscoSystemsSIP-GW-UserAgent 7826 3751 IN IP4 172.18.193.98</td>
</tr>
<tr>
<td>s=SIP Call</td>
<td>s=SIP Call</td>
</tr>
<tr>
<td>c=IN IP4 172.18.193.98</td>
<td>c=IN IP4 172.18.193.98</td>
</tr>
<tr>
<td>t=0 0</td>
<td>t=0 0</td>
</tr>
<tr>
<td>m=audio 51372 RTP/SAVP 0</td>
<td>m=audio 49170 RTP/SAVP 0</td>
</tr>
<tr>
<td>a=rtpmap:0 PCMU/8000</td>
<td>a=rtpmap:0 PCMU/8000</td>
</tr>
<tr>
<td>a=crypto:1 AES_CM_128_HMAC_SHA1_32 inline:D0RmdmcmVCspEc3QGZINWpVLFeHtQXctfHwJSoj</td>
<td>a=crypto:1 AES_CM_128_HMAC_SHA1_32 inline:NzB4d1BINUAsLEx6UzF3WSJhPSdFeGdUJSbpX1Zj</td>
</tr>
</tbody>
</table>

Note

The application is notified of the content in Crypto and inline SDP lines.

Multiple XMF Applications and Recording Tone

Multiple XMF allows multiple (maximum 32) web applications to register with the XMF provider as separate XMF applications and provide redundancy for the voice calls recording. Recording tone provides recording tone capability to the recording sessions. Recording tone is supported for IP to IP, IP to TDM, and TDM to TDM trunks.
An example topology is as shown below where 4 CUCM applications are deployed. CUCM triggers media forking request to Cisco UBE. Recording tone is played to the parties involved in the call based on the recordTone parameter set in the media forking request.

**Figure 3: Multiple XMF Applications and Recording Tone**

Media forking can be invoked using any of the following APIs:

- RequestXmfConnectionMediaForking
- RequestXmfCallMediaForking
- RequestXmfCallMediaSetAttributes

The “recordTone” parameter can be enabled in any of the above requests and recording tone will be played for the parties involved in the call. The “recordTone” parameter in the API request can have the following values:

- COUNTRY_US
- COUNTRY AUSTRALIA
- COUNTRY_GERMANY
- COUNTRY RUSSIA
There is no difference in the recording tone beep when any country value is chosen. Recording tone beep is played at an interval of every 15 seconds. Digital signal processors and other resources are not utilized for playing recording tone even for transcoded calls. No specific configuration is required to enable or disable recording tone. By default, no recording tone is enabled.

If "recordTone" parameter is enabled only on the farEndAddr, then this tone is played only on the outgoing leg. Likewise, if enabled only on the nearEndAddr, then the tone is played only on the incoming leg. When enabled in both the far and near end, then recording tone is played on both the legs.

The RequestXmfConnectionMediaForking API allows insertion of recording tone on a per connection basis. There could be scenarios where one leg receives two recordTone insertion requests. When a leg receives recordTone insertion request, the nearEnd request always takes precedence over the farEnd request.

**Forking Preservation**

After media forking is initiated by the web application, the forking can be preserved to continue the recording, even if the WAN connection to the application is lost or if the application is unregistered.

The "preserve" parameter value can be set to TRUE or FALSE in any of the 3 forking requests (RequestXmfConnectionMediaForking, RequestXmfCallMediaForking, or RequestXmfCallMediaSetAttributes) from the application to Cisco UBE.

- If the "preserve" parameter received is TRUE, then forking will continue the recording, even if the WAN connection to application is lost or application is unregistered.
- If the "preserve" parameter received is FALSE, then forking will not continue the recording.
If the "preserve" parameter is not received in the media forking request, then forking will not continue the recording.

# How to Configure UC Gateway Services

## Configuring Cisco Unified Communication IOS Services on the Device

### SUMMARY STEPS

1. enable
2. configure terminal
3. ip http server
4. ip http max-connections value
5. ip http timeout-policy idle seconds life seconds requests value
6. http client connection idle timeout seconds
7. uc wsapi
8. message-exchange max-failures number
9. probing max-failures number
10. probing interval keepalive seconds
11. probing interval negative seconds
12. source-address ip-address
13. end

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>Enters your password if prompted.</td>
</tr>
<tr>
<td></td>
<td>Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>ip http server</td>
<td>Enables the HTTP server (web server) on the system.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config)# ip http server</td>
<td></td>
</tr>
</tbody>
</table>
### Configuring Cisco Unified Communication IOS Services on the Device

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 4</strong></td>
<td><strong>ip http max-connections value</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;Device(config)# ip http max-connection 100</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td><strong>ip http timeout-policy idle seconds life seconds requests value</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;Device(config)# ip http timeout-policy idle 600 life 864000 requests 86400</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td><strong>http client connection idle timeout seconds</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;Device(config)# http client connection idle timeout 600</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td><strong>uc wsapi</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;Device(config)# uc wsapi</td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
</tr>
<tr>
<td>--------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>Step 8</td>
<td>message-exchange max-failures number</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>Device(config-uc-wsapi)# message-exchange max-failures 2</td>
</tr>
<tr>
<td>Step 9</td>
<td>probing max-failures number</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>Device(config-uc-wsapi)# probing max-failures 5</td>
</tr>
<tr>
<td>Step 10</td>
<td>probing interval keepalive seconds</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>Device(config-uc-wsapi)# probing interval keepalive 255</td>
</tr>
<tr>
<td>Step 11</td>
<td>probing interval negative seconds</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>Device(config-uc-wsapi)# probing interval negative 10</td>
</tr>
<tr>
<td>Step 12</td>
<td>source-address ip-address</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>Device(config-uc-wsapi)# source-address 192.1.12.14</td>
</tr>
<tr>
<td>Step 13</td>
<td>end</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>Device(config-uc-wsapi)# end</td>
</tr>
</tbody>
</table>
Configuring the XMF Provider

SUMMARY STEPS

1. enable
2. configure terminal
3. uc wsapi
4. source-address ip address
5. provider xmf
6. no shutdown
7. remote-url index url
8. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** enable | Enables privileged EXEC mode.  
• Enter your password if prompted. |
<p>| Example: Device&gt; enable | |
| <strong>Step 2</strong> configure terminal | Enters global configuration mode. |
| Example: Device# configure terminal | |
| <strong>Step 3</strong> uc wsapi | Enters Cisco Unified Communication IOS Service configuration mode. |
| Example: Device(config)# uc wsapi | |
| <strong>Step 4</strong> source-address ip address | Configures the source ip address. |
| Example: Device(config)# source-address 172.156.19.38 | |
| <strong>Step 5</strong> provider xmf | Enters XMF provider configuration mode. |
| Example: Device(config-uc-wsapi)# provider xmf | |</p>
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 6</strong></td>
<td>no shutdown</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config-uc-wsapi)# no shutdown</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>remote-url index url</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config-uc-wsapi)# remote-url 1 <a href="http://test.com:8090/ucm_xmf">http://test.com:8090/ucm_xmf</a></td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>end</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config-uc-wsapi)# end</td>
</tr>
</tbody>
</table>

### Verifying the UC Gateway Services

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

**SUMMARY STEPS**

1. enable
2. show wsapi registration all
3. show wsapi registration xmf remote-url-index
4. show call media-forking

**DETAILED STEPS**

**Step 1** enable
Enables privileged EXEC mode.

**Example:**
Device> enable

**Step 2** show wsapi registration all
Displays the details of applications registered. Each registered application is identified by a different ID.
Example:
Device# show wsapi registration all

Provider XMF

registration index: 11
  id: 2E7C3034:XMF:myapp:26
  appUrl: http://pascal-lnx.cisco.com:8094/xmf
  appName: myapp
  provUrl: http://9.45.46.16:8090/cisco_xmf
  prober state: STEADY
  connEventsFilter: CREATED|REDIRECTED|ALERTING|CONNECTED|TRANSFERRED|CALL_DELIVERY|DISCONNECTED|HANDOFF_JOIN|HANDOFF_LEAVE
  mediaEventsFilter: DTMF|MEDIA_ACTIVITY|MODE_CHANGE|TONE_DIAL|TONE_OUT_OF_SERVICE|TONE_SECOND_DIAL

registration index: 1
  id: 2E7C304A:XMF:myapp:27
  appUrl: http://pascal-lnx.cisco.com:8092/xmf
  appName: myapp
  provUrl: http://9.45.46.16:8090/cisco_xmf
  prober state: STEADY
  connEventsFilter: CREATED|REDIRECTED|ALERTING|CONNECTED|TRANSFERRED|CALL_DELIVERY|DISCONNECTED|HANDOFF_JOIN|HANDOFF_LEAVE
  mediaEventsFilter: DTMF|MEDIA_ACTIVITY|MODE_CHANGE|TONE_DIAL|TONE_OUT_OF_SERVICE|TONE_SECOND_DIAL

registration index: 21
  id: 2E7C6423:XMF:myapp:28
  appUrl: http://pascal-lnx.cisco.com:8096/xmf
  appName: myapp
  provUrl: http://9.45.46.16:8090/cisco_xmf
  prober state: STEADY
  connEventsFilter: CREATED|REDIRECTED|ALERTING|CONNECTED|TRANSFERRED|CALL_DELIVERY|DISCONNECTED|HANDOFF_JOIN|HANDOFF_LEAVE
  mediaEventsFilter: DTMF|MEDIA_ACTIVITY|MODE_CHANGE|TONE_DIAL|TONE_OUT_OF_SERVICE|TONE_SECOND_DIAL

registration index: 31
  id: 2E7C69E8:XMF:myapp:29
  appUrl: http://pascal-lnx.cisco.com:8098/xmf
  appName: myapp
  provUrl: http://9.45.46.16:8090/cisco_xmf
  prober state: STEADY
  connEventsFilter: CREATED|REDIRECTED|ALERTING|CONNECTED|TRANSFERRED|CALL_DELIVERY|DISCONNECTED|HANDOFF_JOIN|HANDOFF_LEAVE
  mediaEventsFilter: DTMF|MEDIA_ACTIVITY|MODE_CHANGE|TONE_DIAL|TONE_OUT_OF_SERVICE|TONE_SECOND_DIAL

Step 3
show wsapi registration xmf remote-url-index
Displays the details of only a particular XMF registered application with any ID ranging from 1 to 32.

Example:
Device# show wsapi registration xmf 1

Provider XMF

registration index: 1
  id: 2E7C6423:XMF:myapp:28
  appUrl: http://pascal-lnx.cisco.com:8096/xmf
  appName: myapp
  provUrl: http://9.45.46.16:8090/cisco_xmf
  prober state: STEADY
  connEventsFilter: CREATED|REDIRECTED|ALERTING|CONNECTED|TRANSFERRED|CALL_DELIVERY|DISCONNECTED|HANDOFF_JOIN|HANDOFF_LEAVE
Step 4  show call media-forking
Displays the forked stream information.

Example:
Device# show call media-forking
Warning: Output may be truncated if sessions are added/removed concurrently!

<table>
<thead>
<tr>
<th>Session</th>
<th>Call</th>
<th>n/f</th>
<th>Destination (port address)</th>
</tr>
</thead>
<tbody>
<tr>
<td>187</td>
<td>BA</td>
<td>n/f</td>
<td>45864 10.104.105.232</td>
</tr>
<tr>
<td>188</td>
<td>BA</td>
<td>far</td>
<td>54922 10.104.105.232</td>
</tr>
<tr>
<td>189</td>
<td>B9</td>
<td>near</td>
<td>45864 10.104.105.232</td>
</tr>
<tr>
<td>190</td>
<td>B9</td>
<td>far</td>
<td>54922 10.104.105.232</td>
</tr>
</tbody>
</table>

Troubleshooting Tips

You can use the following debug commands to troubleshoot the UC Gateway Services configurations.

- debug wsapi infrastructure all
- debug wsapi xcc all
- debug wsapi xmf all
- debug wsapi xmf messages
- debug wsapi infrastructure detail
- debug voip application
• debug voip application media forking

Configuration Examples for UC Gateway Services

Example: Configuring Cisco Unified Communication IOS Services

The following example shows how to configure the device for Cisco Unified Communication IOS Services and enable the HTTP server:

```
Device> enable
Device# configure terminal
Device(config)# ip http server
Device(config)# ip http max-connection 100
Device(config)# ip http timeout-policy idle 600 life 86400 requests 86400
Device(config)# http client connection idle timeout 600
Device(config)# uc wsapi
Device(config-uc-wsapi)# message-exchange max-failures 2
Device(config-uc-wsapi)# probing max-failures 5
Device(config-uc-wsapi)# probing interval keepalive 255
Device(config-uc-wsapi)# probing interval negative 10
Device(config-uc-wsapi)# source-address 192.1.12.14
Device(config-uc-wsapi)# end
```

Example: Configuring the XMF Provider

The following example shows how to enable the XMF providers. The configuration specifies the address and port that the application uses to communicate with the XMF provider:

```
Device> enable
Device# configure terminal
Device(config)# uc wsapi
Device(config-uc-wsapi)# provider xmf
Device(config-uc-wsapi)# no shutdown
Device(config-uc-wsapi)# remote-url 1 http://test.com:8090/ucm_xmf
Device(config-uc-wsapi)# end
```

Example: Configuring UC Gateway Services

```
uc wsapi
  message-exchange max-failures 5
  response-timeout 10
  source-address 192.1.12.14
  probing interval negative 20
  probing interval keepalive 250
  !
  provider xmf
  remote-url 1 http://pascal-lnx.cisco.com:8050/ucm_xmf
```
CHAPTER 14

Dynamic Payload Type Interworking for DTMF and Codec Packets for SIP-to-SIP Calls

The Dynamic Payload Type Interworking for DTMF and Codec Packets for SIP-to-SIP Calls feature provides dynamic payload type interworking for dual tone multifrequency (DTMF) and codec packets for Session Initiation Protocol (SIP) to SIP calls.

Based on this feature, the Cisco Unified Border Element (Cisco UBE) interworks between different dynamic payload type values across the call legs for the same codec. Also, Cisco UBE supports any payload type value for audio, video, named signaling events (NSEs), and named telephone events (NTEs) in the dynamic payload type range 96 to 127.

- Feature Information for Dynamic Payload Type Interworking for DTMF and Codec Packets for SIP-to-SIP Calls, page 141
- Restrictions for Dynamic Payload Type Interworking for DTMF and Codec Packets for SIP-to-SIP Calls, page 142
- Symmetric and Asymmetric Calls, page 143
- High Availability Checkpointing Support for Asymmetric Payload, page 144
- How to Configure Dynamic Payload Type Passthrough for DTMF and Codec Packets for SIP-to-SIP Calls, page 144
- Configuration Examples for Assymetric Payload Interworking, page 148

Feature Information for Dynamic Payload Type Interworking for DTMF and Codec Packets for SIP-to-SIP Calls

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 . An account on Cisco.com is not required.
Table 14: Feature Information for Dynamic Payload Interworking for DTMF and Codec Packets Support

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic Payload Type Interworking for DTMF and Codec Packets for SIP-to-SIP Calls</td>
<td>15.0(1)XA 15.1(1)T</td>
<td>The Dynamic Payload Type Interworking for DTMF and Codec Packets for SIP-to-SIP Calls feature provides dynamic payload type interworking for DTMF and codec packets for SIP-to-SIP calls. The following commands were introduced or modified: asymmetric payload and voice-class sip asymmetric payload.</td>
</tr>
<tr>
<td>Dynamic Payload Type Interworking for DTMF and Codec Packets for SIP-to-SIP Calls</td>
<td>Cisco IOS Release XE 3.1S</td>
<td>The Dynamic Payload Type Interworking for DTMF and Codec Packets for SIP-to-SIP Calls feature provides dynamic payload type interworking for DTMF and codec packets for SIP-to-SIP calls. The following commands were introduced or modified: asymmetric payload and voice-class sip asymmetric payload.</td>
</tr>
<tr>
<td>High Availability Checkpointing Support for Asymmetric Payload</td>
<td>15.4(2)T</td>
<td>High availability support for asymmetric payload type interworking was added.</td>
</tr>
</tbody>
</table>

Restrictions for Dynamic Payload Type Interworking for DTMF and Codec Packets for SIP-to-SIP Calls

The Dynamic Payload Type Interworking for DTMF and Codec Packets for SIP-to-SIP Calls feature is not supported for the following:

- H323-to-H323 and H323-to-SIP calls.
- All transcoded calls.
- Secure Real-Time Protocol (SRTP) pass-through calls.
- Flow-around calls.
- Asymmetric payload types are not supported on early-offer (EO) call legs in a delayed-offer to early-offer (DO-EO) scenario.
- Cisco fax relay.
Multiple $m$ lines with the same dynamic payload types, where $m$ is:

\[ m = \text{audio} \text{<media-port1>} \text{RTP/AVP XXX} m = \text{video} \text{<media-port2>} \text{RTP/AVP XXX} \]

**Symmetric and Asymmetric Calls**

Cisco UBE supports dynamic payload type negotiation and interworking for all symmetric and asymmetric payload type combinations. A call leg on Cisco UBE is considered as symmetric or asymmetric based on the payload type value exchanged during the offer and answer with the endpoint:

- A symmetric endpoint accepts and sends the same payload type.
- An asymmetric endpoint can accept and send different payload types.

The Dynamic Payload Type Interworking for DTMF and Codec Packets for SIP-to-SIP Calls feature is enabled by default for a symmetric call. An offer is sent with a payload type based on the dial-peer configuration. The answer is sent with the same payload type as was received in the incoming offer. When the payload type values negotiated during the signaling are different, the Cisco UBE changes the Real-Time Transport Protocol (RTP) payload value in the VoIP to RTP media path.

To support asymmetric call legs, you must enable The Dynamic Payload Type Interworking for DTMF and Codec Packets for SIP-to-SIP Calls feature. The dynamic payload type value is passed across the call legs, and the RTP payload type interworking is not required. The RTP payload type handling is dependent on the endpoint receiving them.
High Availability Checkpointing Support for Asymmetric Payload

High availability for a call involving asymmetric payloads is supported. In case of fail-over from active to stand-by, the asymmetric payload interworking will be continued as new active CUBE passes across the payload type values according to the negotiation and call establishment.

Figure 5: Sample High-Availability Topology

How to Configure Dynamic Payload Type Passthrough for DTMF and Codec Packets for SIP-to-SIP Calls

Configuring Dynamic Payload Type Passthrough at the Global Level

Perform this task to configure the pass through of DTMF or codec payload to the other call leg (instead of performing dynamic payload type interworking) feature at the global level.

SUMMARY STEPS

1. enable
2. configure terminal
3. voice service voip
4. sip
5. asymmetric payload {dtmf | dynamic-codecs | full | system}
6. end
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><strong>enable</strong></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td><strong>Device&gt; enable</strong></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Step 2</td>
<td><strong>configure terminal</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td><strong>Device# configure terminal</strong></td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td><strong>voice service voip</strong></td>
<td>Enters voice service configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td><strong>Device(config)# voice service voip</strong></td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td><strong>sip</strong></td>
<td>Enters voice service SIP configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td><strong>Device(conf-voi-serv)# sip</strong></td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td>**asymmetric payload {dtmf</td>
<td>dynamic-codecs</td>
</tr>
<tr>
<td>Example:</td>
<td><strong>Device(conf-serv-sip)# asymmetric payload full</strong></td>
<td><strong>Note</strong> The dtmf and dynamic-codecs keywords are internally mapped to the full keyword to provide asymmetric payload type support for audio and video codecs, DTMF, and NSEs.</td>
</tr>
<tr>
<td>Step 6</td>
<td><strong>end</strong></td>
<td>Exits voice service SIP configuration mode and enters privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td><strong>Device(conf-serv-sip)# end</strong></td>
<td></td>
</tr>
</tbody>
</table>

## Configuring Dynamic Payload Type Passthrough for a Dial Peer

Perform this task to configure the pass through of DTMF or codec payload to the other call leg (instead of performing dynamic payload type interworking) feature at the dial-peer level.
## SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `dial-peer voice tag voip`
4. `voice-class sip asymmetric payload {dtmf | dynamic-codecs | full | system}`
5. `end`

## DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| Step 1 | `enable` | Enables privileged EXEC mode.  
  * Enter your password if prompted. |
| Example: | Device> enable |
| Step 2 | `configure terminal` | Enters global configuration mode. |
| Example: | Device# configure terminal |
| Step 3 | `dial-peer voice tag voip` | Enters dial peer voice configuration mode. |
| Example: | Device(config)# dial-peer voice 77 voip |
| Step 4 | `voice-class sip asymmetric payload {dtmf | dynamic-codecs | full | system}` | Configures the dynamic SIP asymmetric payload support.  
  **Note** The `dtmf` and `dynamic-codecs` keywords are internally mapped to the `full` keyword to provide asymmetric payload type support for audio and video codecs, DTMF, and NSEs. |
| Example: | Device(config-dial-peer)# voice-class sip asymmetric payload full |
| Step 5 | `end` | *(Optional)* Exits dial peer voice configuration mode and enters privileged EXEC mode. |
| Example: | Device(config-dial-peer)# end |
Verifying Dynamic Payload Interworking for DTMF and Codec Packets Support

This task shows how to display information to verify Dynamic Payload Type Interworking for DTMF and Codec Packets for SIP-to-SIP Calls configuration feature. These `show` commands need not be entered in any specific order.

**SUMMARY STEPS**

1. `enable`
2. `show call active voice compact`
3. `show call active voice`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> show call active voice compact</td>
<td>(Optional) Displays a compact version of call information.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device# show call active voice compact</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> show call active voice</td>
<td>(Optional) Displays call information for voice calls in progress.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device# show call active voice</td>
<td></td>
</tr>
</tbody>
</table>

**Troubleshooting Tips**

Use the following commands to debug any errors that you may encounter when you configure the Dynamic Payload Type Interworking for DTMF and Codec Packets for SIP-to-SIP Calls feature:

- `debug ccsip all`
- `debug voip ccapi inout`
- `debug voip rtp`

Use the following debug commands to troubleshoot HA Checkpointing for Asymmetric Payload:

- `debug voip ccapi all`
• debug voice high-availability all
• debug voip rtp error
• debug voip rtp inout
• debug voip rtp packet
• debug voip rtp high-availability
• debug voip rtp function
• debug ccsip all

Use the following `show` commands to troubleshoot HA Checkpointing for Asymmetric Payload:

• show redundancy state
• show redundancy inter-device
• show standby brief
• show voice high-availability summary
• show voip rtp stats
• show voip rtp high-availability stats
• show voip rtp connection detail
• show call active voice brief
• show call active voice [summary]
• show call active video brief
• show call active video [summary]
• show align
• show memory debug leak

Configuration Examples for Assymetric Payload Interworking

Example: Asymmetric Payload Interworking—Passthrough Configuration

```plaintext
! voice service voip
   allow-connections sip to sip
   sip
     rel1xx disable
     asymmetric payload full
     midcall-signaling passthru
     !
   dial-peer voice 1 voip
     voice-class sip asymmetric payload full
     session protocol sipv2
     rtp payload-type cisco-codec-fax-ind 110
     rtp payload-type cisco-codec-video-h264 112
```
Example: Asymmetric Payload Interworking—Interworking Configuration

```plaintext
session target ipv4:9.13.8.23

Example: Asymmetric Payload Interworking—Interworking Configuration

! voice service voip
  allow-connections sip to sip
! dial-peer voice 1 voip
  session protocol sipv2
  rtp payload-type cisco-codec-fax-ind 110
  rtp payload-type cisco-codec-video-h264 112
  session target ipv4:9.13.8.23
```
Example: Asymmetric Payload Interworking—Interworking Configuration
Acoustic Shock Protection

Acoustic Shock Protection (ASP) is a voice circuit-breaker feature that is designed to protect users, especially those wearing headsets, from exposure to loud, sustained, and piercing tones, such as those produced by a fax machine. It is a workplace-safety feature for voice calls. When the tone is present at the input of the ASP module, the audio path in the affected direction is muted to protect the listener, and a gentle alert tone is played out for as long as the tone persists. ASP may be inserted in either or both directions of a call, that is, applied to incoming packets to protect the ears of a listener on the Time-Division Multiplexing (TDM) gateway, applied to incoming PSTN calls (microphone signal) to protect the ears of listeners at the other end of the call, or applied to both simultaneously.

- Finding Feature Information, page 151
- Restrictions for ASP, page 151
- Information About ASP, page 152
- How to Configure ASP, page 153
- Configuration Examples for the Acoustic Shock Protection Feature, page 158
- Feature Information for Acoustic Shock Protection, page 159

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for ASP

- Supported on PVDM3 only.
- Supported only on flex codec complexity.
• No support for H.32x video call, complex forking calls, and fax and modem calls.
• No support for TDM hairpin call.
• The configuration under dial peer has higher priority than the configuration at the global level.
• No support for conference calls, IP/SIP phones, and the Skinny Client Control Protocol (SCCP).
• CLI supports enabling ASP but not disabling ASP.
• No support for dynamically enabling or disabling ASP during a call.

Information About ASP

Acoustic Shock Protection

Acoustic Shock Protection (ASP) is an adaptive signal processing algorithm on the Digital Signal Processor (DSP) that analyzes incoming audio for the presence of offending tones that might harm humans. Offending tones include signals that are:

• Loud
• Tonal (energy concentrated around a single frequency)
• Persistent (lasts longer than a few tens of milliseconds)

If an offending tone is present, the audio path in that direction is muted temporarily, and a quiet, alerting signal is played out to the listener side. The call is never dropped; only the audio is muted temporarily. If or when the tone disappears from the input, the mute is removed. ASP does not disrupt low-frequency tones (below 650 Hz) such as ringback, dial, and so forth. Since ASP is designed to mute only single-frequency tones, it allows multi-tone signals such as Dual Tone Multi-Frequency (DTMF) to pass unhindered. ASP is supported on TDM gateways (TDM-VoIP and TDM-TDM) and on the Cisco Unified Border Element (Cisco UBE).

Note

ASP is for voice calls only and not for faxes and modems.

Some of the best practices for ASP are as follows:

• Use default values
• Use ASP on dial peers where you are certain that people (not faxes) are listening.
• Do not use ASP on dial peers associated with fax machines, modems, or TTY/TDD devices. Use fax-relay or modem-relay modes on dial peers dedicated to such devices.
• ASP is designed for deployment in situations where customers have experienced acoustic shock safety issues. If there are issues like false triggering (for example, ASP alerts on regular voices), then you must turn off ASP. You can choose from three detector sensitivity modes: slow, auto, or fast. Fast mode is a highly sensitive hair-trigger. Auto mode is recommended. Slow mode lets more tone leak through, but has better rejection of false triggers.
How to Configure ASP

Creating the Media Profile for ASP

Perform this task to create a media profile to configure acoustic shock protection.

SUMMARY STEPS

1. enable
2. configure terminal
3. media profile asp tag
4. mode mode
5. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example: Device&gt; enable</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> media profile asp tag</td>
<td>Creates the media profile to configure ASP and enters media profile</td>
</tr>
<tr>
<td>Example: Device(config)# media profile asp 5</td>
<td>configuration mode. The range for the media profile tag is from 1 to 10000.</td>
</tr>
<tr>
<td><strong>Step 4</strong> mode mode</td>
<td>Sets the ASP sensitivity mode to preset = auto (which is default). Auto</td>
</tr>
<tr>
<td>Example: Device(cfg-mediaprofile)# mode auto</td>
<td>mode provides a good tradeoff between ASP speed and false trigger</td>
</tr>
<tr>
<td></td>
<td>rejection.</td>
</tr>
<tr>
<td></td>
<td>The other modes are:</td>
</tr>
<tr>
<td></td>
<td>• slow—Presets ASP sensitivity mode to 1. This mode provides slower</td>
</tr>
<tr>
<td></td>
<td>detection speed for reduced chance of false triggers.</td>
</tr>
<tr>
<td></td>
<td>• fast—Presets ASP sensitivity mode to 2. This mode provides faster</td>
</tr>
<tr>
<td></td>
<td>detection speed but higher chance of false triggers.</td>
</tr>
<tr>
<td></td>
<td>• expert—This mode exposes direct control of individual ASP</td>
</tr>
<tr>
<td></td>
<td>parameters and is recommended for test use only.</td>
</tr>
</tbody>
</table>
### Creating the Media Profile to Enable ASP

After the media profile is created, you must create a media class to enable acoustic shock protection. Perform this task to create a media class.

**SUMMARY STEPS**

1. `enable`
2. `configure terminal`
3. `media class` *tag*
4. `asp profile` *tag*
5. `end`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>enable</code></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Device&gt; enable</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
</tbody>
</table>

| **Step 2** `configure terminal` | Enters global configuration mode.                                        |
| **Example:**                 |                                                                         |
| `Device# configure terminal` |                                                                         |

| **Step 3** `media class` *tag* | Creates the media class to enable the acoustic shock protection feature and enters media class configuration mode. The range for the media class tag is from 1 to 10000. |
| **Example:**                   |                                                                         |
| `Device(config)# media class 2` |                                                                         |
### Configuring the Media Class at a Dial Peer Level for ASP

#### SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `dial-peer voice tag pots`
4. `media-class tag`
5. `end`

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>enable</code></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example: Device&gt; enable</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>dial-peer voice tag pots</code></td>
<td>Defines a particular dial peer and enters dial-peer voice</td>
</tr>
<tr>
<td>Example: Device(config)# dial-peer voice 20 pots</td>
<td>configuration mode. The range for the dial-peer voice tag is from 1 to 1073741823.</td>
</tr>
</tbody>
</table>
### Configuring the Media Class Globally for ASP

**SUMMARY STEPS**

1. **enable**
2. **configure terminal**
3. **media service**
4. **enhancement**
5. **tdm tag**
6. **end**

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>enable</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device&gt; enable</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><strong>configure terminal</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device# configure terminal</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><strong>media service</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config)# media service</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td></td>
</tr>
<tr>
<td>enhancement</td>
<td>Enters the submode enhance of media service.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device(cfg-mediaservice)# enhancement</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td></td>
</tr>
<tr>
<td>tdm tag</td>
<td>Applies the TDM call globally. The range for the media class tag number is from 1 to 10000.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device(cfg-service-enhance)# tdm 2</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td></td>
</tr>
<tr>
<td>end</td>
<td>Returns to privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device(config-dial-peer)# end</td>
<td></td>
</tr>
</tbody>
</table>

**Verifying ASP**

Perform this task to verify the voice quality metrics.

**SUMMARY STEPS**

1. enable
2. show call active voice stats | b pid:

**DETAILED STEPS**

**Step 1** enable

**Example:**
Device> enable

Enables privileged EXEC mode.

**Step 2** show call active voice stats | b pid:

**Example:**
Device# show call active voice stats | b pid:1300

11EC : 5 09:14:25.971 PDT Thu Jul 28 2011.1 +1130 pid:1300 Answer 1300 active dur 00:01:36 tx:17/321
rx:17/321 dscp:0 media:0
DSP/TX: PK=17, SG=0, NS=1, DU=90570, VO=320
DSP/RX: PK=17, SG=0, CF=1, RX=90570, VO=320, BS=0, BP=0, LP=0, EP=0
....

---

Cisco Unified Border Element Protocol-Independent Features and Setup Configuration Guide, Cisco IOS Release 15M&T
DSP/DL: RT=0, ED=0
MIC Direction: DSP/NR: NR=1, ND=0, LV=257, IN=1, PN=0, ON=0
DSP/AS: AE=1, AD=0, AV=0, AM=0, NT=0, TT=0, TD=0, LF=0, LD=0
EAR Direction: DSP/NR: NR=0, ND=0, LV=0, IN=0, PN=0, ON=0
DSP/AS: AE=0, AD=0, AV=0, AM=0, NT=0, TT=0, TD=0, LF=0, LD=0
Telephony call-legs: 1
SIP call-legs: 0
H323 call-legs: 1

Displays information about digital signal processing (DSP) voice quality metrics.

**Troubleshooting Tips**

The following commands can help troubleshoot ASP:

- debug voip hpi all
- debug voip dsmp all
- debug voip dsm all
- debug voip vtsp all
- debug vpm dsp all

**Configuration Examples for the Acoustic Shock Protection Feature**

**Example: Enabling ASP Globally**

```
media profile asp 6
!
media class 1
  asp profile 6
!
media service
  enhancement
tdm 1
```

**Example: Enabling ASP on a Dial Peer**

```
media profile asp 4
!
media class 1
  asp profile 4
!
dial-peer voice 2100 pots
destination-pattern 2100
  incoming called-number 1100
```
Feature Information for Acoustic Shock Protection

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 . An account on Cisco.com is not required.

Table 15: Feature Information for Acoustic Shock Protection

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acoustic Shock Protection</td>
<td>15.2(2)T, 15.2(3)T</td>
<td>Acoustic Shock Protection (ASP) is a voice circuit-breaker feature that is designed to protect users, especially those wearing headsets, from exposure to loud, sustained, and piercing tones, such as those produced by a fax machine. It is a workplace-safety feature for voice calls. ASP is supported on TDM gateways and on Cisco UBE. The following commands were introduced or modified: media profile asp, media service.</td>
</tr>
<tr>
<td>Acoustic Shock Protection</td>
<td>Cisco IOS XE Release 3.6S</td>
<td>Acoustic Shock Protection (ASP) is a voice circuit-breaker feature that is designed to protect users, especially those wearing headsets, from exposure to loud, sustained, and piercing tones, such as those produced by a fax machine. It is a workplace-safety feature for voice calls. ASP is supported on TDM gateways and on Cisco UBE. In Cisco IOS XE Release 3.6S, this feature was implemented on the Cisco Unified Border Element (Enterprise) The following commands were introduced or modified: media profile asp, media service.</td>
</tr>
</tbody>
</table>
Noise Reduction

Noise Reduction (NR) is a voice enhancement process that improves the quality of incoming speech that has already been corrupted with background noise; for example, a voice conference participant speaking on a cell-phone in a car. NR works best with steady state broadband noises like engine noise but not as well with impulsive noises like nearby chatter.

- Finding Feature Information, page 161
- Prerequisites for Noise Reduction, page 161
- Restrictions for NR, page 162
- Information About NR, page 162
- How to Configure NR, page 163
- Configuration Examples for the NR feature, page 168
- Feature Information for Noise Reduction, page 169

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Noise Reduction

Cisco Unified Border Element

- Cisco IOS Release 15.2(2)T, or a later release must be installed and running on your Cisco Unified Border Element.
Cisco Unified Border Element (Enterprise)

- Cisco IOS XE Release 3.6S or a later release must be installed and running on your Cisco ASR 1000 Series Router.

Restrictions for NR

- Supported only on PVDM3.
- Supported only on flex codec complexity.
- No support for H.32x video call, complex forking calls, and fax and modem calls.
- No support for Time-Division Multiplexing (TDM) hairpin call.
- Configurations under POTS dial peer has higher priority over VoIP dial peer for NR.
- Configurations under the dial peer has higher priority than configurations at the global level.
- No support for conference calls, IP/SIP phones, and the Skinny Client Control Protocol (SCCP).
- CLI supports enabling NR but not disabling NR.
- No support for dynamically enabling or disabling NR during a call.

Information About NR

Noise Reduction

Noise Reduction (NR) is an adaptive signal processing algorithm on the Digital Signal Processor (DSP) that analyzes incoming audio, extracts a fingerprint of the background noise during talker pauses, and then performs ongoing spectral subtraction of this noise after a short training period (a few seconds). NR constantly adapts to changes in background noises over time.

NR can affect music on hold signals by making the music quieter. NR may disrupt fax/modem/TDD devices, although it is designed to self-disable in those cases. Use modem-relay mode for reliable fax/modem transmission. NR is supported on TDM gateways (TDM-VoIP and TDM-TDM) and on the Cisco Unified Border Element (Cisco UBE).

Some of the best practices for NR are as follows:

- Use default values.
- Do not use NR on dial peers associated with fax machines. Use fax or modem-relay modes for those dial peers.
- NR, when used without dynamic user control of intensity (as is the case with gateways), must be used at a low intensity (default or lower) since it is always on. High intensity is dramatic for demonstrations with loud background noises, but the NR process itself will degrade "normal" calls if NR is run at high intensity.
How to Configure NR

Creating the Media Profile for NR

Perform this task to create a media profile to configure noise reduction parameters.

SUMMARY STEPS

1. enable
2. configure terminal
3. media profile nr tag
4. intensity level
5. noisefloor level
6. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td>• Enter your password if prompted.</td>
<td></td>
</tr>
<tr>
<td>Step 2 configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 3 media profile nr tag</td>
<td>Creates the media profile to configure noise reduction parameters and</td>
</tr>
<tr>
<td>Example:</td>
<td>enters media profile configuration mode. The range for the media</td>
</tr>
<tr>
<td>Device(config)# media profile nr 2</td>
<td>profile tag is from 1 to 10000.</td>
</tr>
<tr>
<td>Step 4 intensity level</td>
<td>Configures the intensity level or depth of the noise reduction process.</td>
</tr>
<tr>
<td>Example:</td>
<td>The range is from 0 to 6.</td>
</tr>
<tr>
<td>Device(cfg-mediaprofile)# intensity 2</td>
<td></td>
</tr>
<tr>
<td>Step 5 noisefloor level</td>
<td>Configures the noise level, in dBm, above which NR will operate.</td>
</tr>
<tr>
<td>Example:</td>
<td>NR will allow noises quieter than this level to pass without processing.</td>
</tr>
<tr>
<td>Device(cfg-mediaprofile)# noisefloor -50</td>
<td></td>
</tr>
</tbody>
</table>
Creating the Media Class to Enable NR

After the media profile is created, you must create a media class to enable noise reduction. Perform this task to create a media class.

SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `media class tag`
4. `nr profile tag`
5. `end`

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device&gt; enable</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> media class tag</td>
<td>Creates the media class to enable the noise reduction feature and enters media class configuration mode. The range for the media class tag is from 1 to 10000.</td>
</tr>
</tbody>
</table>
Configuring the Media Class at a Dial Peer Level for NR

Perform this task to configure the media class for a dial peer.

SUMMARY STEPS

1. enable
2. configure terminal
3. dial-peer voice tag pots
4. media-class tag
5. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example: Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Step 2 configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 3 dial-peer voice tag pots</td>
<td>Defines a particular dial peer and enters the dial-peer voice configuration mode. The range for the dial-peer voice tag is from 1 to 1073741823.</td>
</tr>
<tr>
<td>Example: Device(config)# dial-peer voice 20 pots</td>
<td></td>
</tr>
</tbody>
</table>
Purpose

Command or Action | Purpose
--- | ---

**Step 4**

<table>
<thead>
<tr>
<th>media-class</th>
<th>tag</th>
</tr>
</thead>
</table>

Example:

Device(config-dial-peer)# media-class 2

Applies the media class to the specific dial peer. The range for the media class tag number is from 1 to 10000.

**Step 5**

<table>
<thead>
<tr>
<th>end</th>
</tr>
</thead>
</table>

Example:

Device(config-dial-peer)# end

Returns to the privileged EXEC mode.

## Configuring the Media Class Globally for NR

Perform this task to configure a media class globally.

### SUMMARY STEPS

1. enable
2. configure terminal
3. media service
4. enhancement
5. tdm tag
6. end

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>

**Step 1**

<table>
<thead>
<tr>
<th>enable</th>
</tr>
</thead>
</table>

Example:

Device> enable

Enables privileged EXEC mode.

- Enter your password if prompted.

**Step 2**

<table>
<thead>
<tr>
<th>configure terminal</th>
</tr>
</thead>
</table>

Example:

Device# configure terminal

Enters global configuration mode.
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 3 media service</td>
<td>Enters media service configuration mode.</td>
</tr>
<tr>
<td>Example: Device(config)# media service</td>
<td></td>
</tr>
<tr>
<td>Step 4 enhancement</td>
<td>Enters the submode enhance of media service.</td>
</tr>
<tr>
<td>Example: Device(cfg-mediaservice)# enhancement</td>
<td></td>
</tr>
<tr>
<td>Step 5 tdm tag</td>
<td>Applies the TDM call globally. The range for the media class tag number is from 1 to 10000.</td>
</tr>
<tr>
<td>Example: Device(cfg-service-enhance)# tdm 2</td>
<td></td>
</tr>
<tr>
<td>Step 6 end</td>
<td>Returns to the privileged EXEC mode.</td>
</tr>
<tr>
<td>Example: Device(config-dial-peer)# end</td>
<td></td>
</tr>
</tbody>
</table>

### Verifying NR

Perform this task to verify the voice quality metrics.

#### SUMMARY STEPS

1. enable
2. show call active voice stats | b pid:

#### DETAILED STEPS

**Step 1** enable

Example: Device> enable

Enables privileged EXEC mode.

**Step 2** show call active voice stats | b pid:
Examples:
Device# show call active voice stats | b pid:1300

11EC : 5 09:14:25.971 PDT Thu Jul 28 2011.1 +1130 pid:1300 Answer 1300 active dur 00:01:36 tx:17/321
rx:17/321 dscp:0 media:0
DSP/TX: PK=17, SG=0, NS=1, DU=90570, VO=320
DSP/RX: PK=17, SG=0, CF=1, RX=90570, VO=320, BS=0, BP=0, LP=0, EP=0
....
DSP/DL: RT=0, ED=0
MIC Direction:
DSP/NR: NR=1, ND=0, LV=257, IN=1, PN=0, ON=0
DSP/AS: AE=1, AD=0, AV=0, AM=0, NT=0, DT=0, TD=0, LP=0, LD=0
EAR Direction:
DSP/NR: NR=0, ND=0, LV=0, IN=0, PN=0, ON=0
DSP/AS: AE=0, AD=0, AV=0, AM=0, NT=0, DT=0, TD=0, LP=0, LD=0
11EC : 6 09:14:25.973 PDT Thu Jul 28 2011.2 +1130 pid:2300 Originate 2300 active dur 00:01:36 tx:17/457
rx:17/321 dscp:0 media:0
Telephony call-legs: 1
SIP call-legs: 0
H323 call-legs: 1

Displays information about digital signal processing (DSP) voice quality metrics.

Troubleshooting Tips

The following commands can help troubleshoot NR:

- `debug voip hpi all`
- `debug voip dsmp all`
- `debug voip dsm all`
- `debug voip vtsp all`
- `debug vpm dsp all`

Configuration Examples for the NR feature

Example: Enabling NR globally

```plaintext
media profile nr 1
  intensity 1
!
media profile nr 2
!
media profile nr 3
  intensity 2
!
media profile nr 4
  intensity 3
!
media profile nr 5
  intensity 2
```

Cisco Unified Border Element Protocol-Independent Features and Setup Configuration Guide, Cisco IOS Release 15M&T

168
Example: Enabling NR on a Dial Peer

```
media profile nr 1
intensity 1
! media profile nr 2
intensity 2
! media profile nr 3
intensity 2
! media profile asp 4
!
media class 1
nr profile 2
asp profile 4
!
dial-peer voice 2100 pots
destination-pattern 2100
incoming called-number 1100
media-class 1
port 0/2/0:1
forward-digits all

dial-peer voice 1300 voip
destination-pattern 1300
session target ipv4:1.2.146.102
media-class 1
```
### Table 16: Feature Information for Noise Reduction

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noise Reduction</td>
<td>15.2(2)T, 15.2(3)T</td>
<td>Noise Reduction (NR) is a voice enhancement or restoration process that improves the quality of incoming speech that has already been corrupted with background noise. NR is supported on TDM gateways and on the Cisco UBE. The following commands were introduced or modified: <code>intensity</code>, <code>media profile nr</code>, <code>media service</code>, and <code>noisefloor</code>.</td>
</tr>
<tr>
<td>Noise Reduction</td>
<td>Cisco IOS XE Release 3.6S</td>
<td>Noise Reduction (NR) is a voice enhancement or restoration process that improves the quality of incoming speech that has already been corrupted with background noise. NR is supported on TDM gateways and on Cisco UBE. In Cisco IOS XE Release 3.6S, this feature was implemented on the Cisco Unified Border Element (Enterprise). The following commands were introduced or modified: <code>intensity</code>, <code>media profile nr</code>, <code>media service</code>, <code>noisefloor</code>.</td>
</tr>
</tbody>
</table>
iLBC Support for SIP and H.323

The internet Low Bitrate Codec (iLBC) is a standard, high-complexity speech codec suitable for robust voice communication over IP. The iLBC has built-in error correction functionality that helps the codec perform in networks with high-packet loss. This codec is supported on both Session Initiation Protocol (SIP) and H.323.

- Finding Feature Information, page 171
- Prerequisites for iLBC Support for SIP and H.323, page 171
- Restrictions for iLBC Support for SIP and H.323, page 172
- Information About iLBC Support for SIP and H.323, page 172
- How to Configure an iLBC Codec, page 172
- Feature Information for iLBC Support for SIP and H.323, page 176

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for iLBC Support for SIP and H.323

Cisco Unified Border Element

- Cisco IOS Release 12.2(11)T or a later release must be installed and running on your Cisco Unified Border Element.
Cisco Unified Border Element (Enterprise)

- Cisco IOS XE Release 2.5 or a later release must be installed and running on your Cisco ASR 1000 Series Router.

Restrictions for iLBC Support for SIP and H.323

The iLBC Support for SIP and H.323 feature is supported on the following:

- IP-to-IP gateways with no transcoding and conferencing
- All c5510 DSP-based platforms

Information About iLBC Support for SIP and H.323

The internet Low Bit Rate Codec (iLBC) is designed for narrow band speech and results in a payload bit rate of 13.33 kbits per second for 30-millisecond (ms) frames and 15.20 kbits per second for 20 ms frames.

When the codec operates at block lengths of 20 ms, it produces 304 bits per block, which is packetized as defined in RFC 3952. Similarly, for block lengths of 30 ms it produces 400 bits per block, which is packetized as defined in RFC 3952.

The iLBC has built-in error correction functionality to provide better performance in networks with higher packet loss.

How to Configure an iLBC Codec

Configuring an iLBC Codec on a Dial Peer

The iLBC is intended for packet-based communication. Perform the following steps to configure the iLBC codec on a dial peer.

SUMMARY STEPS

1. enable
2. configure terminal
3. dial-peer voice tag voip
4. rtp payload-type cisco-codec-ilbc [number
5. codec ilbc [mode frame_size [bytes payload_size]]
6. exit
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** | enable | Enables privileged EXEC mode.  
  - Enter your password if prompted.  
  |  
  **Example:** | Device> enable |
| **Step 2** | configure terminal | Enters global configuration mode.  
  |  
  **Example:** | Device# configure terminal |
| **Step 3** | dial-peer voice *tag* voip | Enters dial-peer configuration mode for the VoIP dial peer designated by *tag*.  
  |  
  **Example:** | Device(config)# dial-peer voice 10 voip |
| **Step 4** | rtp payload-type cisco-codec-ilbc *number* | Identifies the payload type of a Real-Time Transport Protocol (RTP) packet.  
  Keyword and argument are as follows:  
  - *cisco-codec-ilbc* [number]--Payload type is for internet Low Bit Rate Codec (iLBC). Range: 96 to 127. Default: 116.  
  **Note** Do not use the following numbers because they have preassigned values: 96, 97, 100, 117, 121 to 123, and 125 to 127. If you use these values, the command will fail. You must first reassign the value in use to a different unassigned number, for example:  
    - rtp payload-type nse 105  
    - rtp payload-type cisco-codec-ilbc 100  
  |  
  **Example:** | Device(config-dial-peer)# rtp payload-type cisco-codec-ilbc 100 |
| **Step 5** | codec ilbc *mode frame_size [bytes payload_size]* | Specifies the voice coder rate of speech for a dial peer. Keywords and arguments are as follows:  
  - *mode* frame_size--The iLBC operating frame mode that will be encapsulated in each packet. Valid entries are 20 (20ms frames for 15.2kbps bit rate) or 30 (30ms frames for 13.33 kbps bit rate). Default is 20.  
    - *bytes* payload_size--Number of bytes in an RTP packet. For mode 20, valid values are 38 (default), 76, 114, 152, 190, and 228. For mode 30, valid values are 50 (default), 100, 150, and 200.  
  |  
  **Example:** | Device(config-dial-peer)# codec ilbc mode 30 bytes 200 |
| **Step 6** | exit | Exits the current mode.  
  |  
  **Example:** | Device(config-dial-peer)# exit |
Configuring an iLBC Codec in the Voice Class

When using multiple codecs, you must create a voice class in which you define a selection order for codecs; then, you can apply the voice class to VoIP dial peers. The `voice class codec` global configuration command allows you to define the voice class that contains the codec selection order. Then, use the `voice-class codec` dial-peer configuration command to apply the class to individual dial peers.

To configure an iLBC in the voice class for multiple-codec selection order, perform the following steps.

You can configure more than one voice class codec list for your network. Configure the codec lists and apply them to one or more dial peers based on which codecs (and the order) you want supported for the dial peers. Define a selection order if you want more than one codec supported for a given dial peer.

SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `voice class codec tag`
4. `codec preference value ilbc [mode frame_size] [bytes payload_size]`
5. `exit`
6. `dial-peer voice tag voip`
7. `voice-class codec tag`
8. `exit`

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>enable</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device&gt; enable</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>configure terminal</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device# configure terminal</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>voice class codec tag</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config)# voice class codec 99</td>
</tr>
<tr>
<td></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Enters voice-class configuration mode and assigns an identification tag number for a codec voice class. The argument is as follows:</td>
</tr>
<tr>
<td></td>
<td>• <code>tag</code> --Unique identifier on the router. Range is 1 to 10000.</td>
</tr>
<tr>
<td><strong>Command or Action</strong></td>
<td><strong>Purpose</strong></td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| **Step 4** codec preference value ilbc [mode frame_size] [bytes payload_size] | Specifies a list of preferred codecs to use on a dial peer. Keywords and arguments are as follows:  
  - `value` -- Order of preference, with 1 being the most preferred and 14 being the least preferred.  
  - `mode frame_size` -- The iLBC operating frame mode that will be encapsulated in each packet. Valid entries are 20 (20ms frames for 15.2kbps bit rate) or 30 (30ms frames for 13.33 kbps bit rate). Default is 20.  
  - `bytes payload_size` -- Number of bytes in an RTP packet. For mode 20, valid values are 38 (default), 76, 114, 152, 190, and 228. For mode 30, valid values are 50 (default), 100, 150, and 200. |
| **Example:** Device(config-voice-class)# codec preference 1 ilbc 30 200 | |
| **Step 5** exit | Exits the current mode. |
| **Example:** Device(config-voice-class)# exit | |
| **Step 6** dial-peer voice tag voip | Enters dial-peer configuration mode for the specified VoIP dial peer. |
| **Example:** Device(config)# dial-peer voice 16 voip | |
| **Step 7** voice-class codec tag | Assigns a previously configured codec selection preference list (the codec voice class that you defined in step 3) to the specified VoIP dial peer.  
  **Note** The `voice-class codec` command in dial-peer configuration mode contains a hyphen. The `voice class` command in global configuration mode does not contain a hyphen. |
| **Example:** Device(config-dial-peer)# voice-class codec 99 | |
| **Step 8** exit | Exits the current mode. |
| **Example:** Device(config-dial-peer)# exit | |

**Verifying iLBC Support for SIP and H.323**

You can use the following commands to check iLBC status:

- `show voice call summary`
- `show voice call status`
Feature Information for iLBC Support for SIP and H.323

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 . An account on Cisco.com is not required.

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
</table>
| iLBC Support for SIP and H.323 | 12.2(11)T 12.2(15)T | The iLBC is a standard, high-complexity speech codec suitable for robust voice communication over IP. The iLBC has built-in error correction functionality that helps the codec perform in networks with high-packet loss. This codec is supported on both Session Initiation Protocol (SIP) and H.323.

The following commands were introduced or modified: `codec ilbc`, `codec preference`, and `rtp payload-type`. |
<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>iLBC Support for SIP and H.323</td>
<td>Cisco IOS XE Release 2.5</td>
<td>The iLBC is a standard, high-complexity speech codec suitable for robust voice communication over IP. The iLBC has built-in error correction functionality that helps the codec perform in networks with high-packet loss. This codec is supported on both Session Initiation Protocol (SIP) and H.323. The following commands were introduced or modified: <code>codec ilbc</code>, <code>codec preference</code>, and <code>rtp payload-type</code>.</td>
</tr>
</tbody>
</table>
CHAPTER 18

Configuring RTP Media Loopback for SIP Calls

RTP packets are looped back toward the source device when the RTP Media Loopback for SIP Calls feature is configured on a dial peer. The SIP RTP media loopback can be used during Cisco UBE deployments to make test calls to verify the media path between the endpoints and Cisco UBE. In a voice loopback call, an echo is heard at the device originating the call. In a video loopback call, the locally captured video and the audio echo must be rendered at the source device.

- Finding Feature Information, page 179
- Prerequisites, page 179
- Restrictions, page 180
- Information About RTP Media Loopback for SIP Calls, page 180
- How to Configure RTP Media Loopback for SIP Calls, page 180
- Configuration Examples for RTP Media Loopback, page 182
- Feature Information for RTP Media Loopback for SIP Calls, page 183

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites

- Media packets must be enabled to pass through the gateway.

- Use the media flow-through command in dial peer voice or voice service configuration mode to enable the media packets.
Cisco Unified Border Element

- Cisco IOS Release 15.1(4)M or a later release must be installed and running on your Cisco Unified Border Element.

Cisco Unified Border Element (Enterprise)

- Cisco IOS XE Release 3.3S or a later release must be installed and running on your Cisco ASR 1000 Series Router.

Restrictions

- SRTP, DTLS, and STUN are not supported in loopback mode.
- Fax (midcall transmit function change) is not supported.
- RSVP is not supported.
- Call transfer is not supported.

Information About RTP Media Loopback for SIP Calls

Digital Signal Processors (DSP) generate and transmit Real-time Transport Protocol (RTP) media packets from a source to a destination transport address during a SIP call session. However, when a SIP call is put on hold the DSP stops generating the RTP media packets and resumes generating and transmitting these media packets after the SIP call is resumed. This ensures that the RTP sequence number is continuous from the time of the origin to the end of a SIP call.

How to Configure RTP Media Loopback for SIP Calls

RTP packets are looped back toward the source device when the RTP Media Loopback for SIP Calls feature is configured on a dial peer. Perform this task to enable the RTP Media Loopback for SIP Calls feature on a dial peer.

SUMMARY STEPS

1. enable
2. configure terminal
3. dial-peer voice tag voip
4. destination-pattern string
5. session protocol sipv2
6. session target loopback:rtp
7. incoming called-number string
8. exit
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>enable</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Specifies that the dial peer is a VoIP peer and enters dial peer voice configuration mode.</td>
</tr>
<tr>
<td>dial-peer voice tag voip</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# dial-peer voice 77 voip</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Specifies the prefix or the full E.164 number for the dial peer.</td>
</tr>
<tr>
<td>destination-pattern string</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-dial-peer)# destination-pattern 77</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Specifies the session protocol for calls with the SIP option.</td>
</tr>
<tr>
<td>session protocol sipv2</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-dial-peer)# session protocol sipv2</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>Designates a network-specific address to receive calls from a VoIP dial peer and configures all voice data to loop back to the source.</td>
</tr>
<tr>
<td>session target loopback:rtp</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-dial-peer)# session target loopback:rtp</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>Specifies a digit string that can be matched by an incoming call to associate the call with the dial peer.</td>
</tr>
<tr>
<td>incoming called-number string</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-dial-peer)# incoming called-number 77</td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>Exits dial peer voice configuration mode and enters global configuration mode.</td>
</tr>
<tr>
<td>exit</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-dial-peer)# exit</td>
<td></td>
</tr>
</tbody>
</table>
Configuration Examples for RTP Media Loopback

Example: Configuring Video Loopback with Cisco Telepresence System

The following sample output shows Media Loopback for SIP Calls configured on a Cisco Telepresence System (CTS).

```
! codec profile 1 aacld
   ftmp "ftmp:96
      profile-level-id=16;streamtype=5;mode=AAChbr;config=B98C00;sizeLength=13;indexLength=3;indexDeltaLength=3;constantDuration=480"
!
! codec profile 2 h264
   ftmp "ftmp:112 profile-level-id=4D0028;sprop-parametersets=R00ARKmWUgDwBDyA,SGE7jyA~packetization-mode=1"
!
! voice class codec 4
   codec preference 1 aacld profile 1
   video codec h264 profile 2
!
! dial-peer voice 2000 voip
   destination-pattern 2000
   rtp payload-type cisco-codec-fax-ind 110
   rtp payload-type cisco-codec-aacld 96
   rtp payload-type cisco-codec-video-h264 112
   session protocol sipv2
   session target loopback:rtp
   incoming called-number 2000
   voice-class codec 4
   voice-class sip bandwidth audio tias-modifier 64000
   voice-class sip bandwidth video tias-modifier 4500000
```

Example: Configuring Video Loopback with Cisco Unified Video Advantage

The following sample output shows Media Loopback for SIP Calls configured on a Cisco Unified Video Advantage (CUVA).

```
! codec profile 3 h264
   ftmp "ftmp:98 profile-level-id=420015"
!
! voice class codec 6
   codec preference 1 g711ulaw
   video codec h264 profile 3
!
! dial-peer voice 5000 voip
   destination-pattern 5000
   rtp payload-type cisco-codec-video-h264 98
   session protocol sipv2
   session target loopback:rtp
   incoming called-number 5000
   voice-class codec 6
   voice-class sip bandwidth video tias-modifier 384000
```
# Feature Information for RTP Media Loopback for SIP Calls

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 . An account on Cisco.com is not required.

Feature History Table entry for the Cisco Unified Border Element.

## Table 18: Feature Information for RTP Media Loopback for SIP Calls

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTP Media Loopback for SIP Calls</td>
<td>15.1(4)M, 15.2(1)T</td>
<td>RTP packets are looped back toward the source when the RTP Media Loopback for SIP Calls feature is configured on a dial peer. SIP RTP media loopback helps in verifying the media path between the device originating the call and the intermediate device. The following commands were introduced or modified: None.</td>
</tr>
</tbody>
</table>
SIP Ability to Send a SIP Registration Message on a Border Element

- Finding Feature Information, page 185
- Prerequisites for SIP Ability to Send a SIP Registration Message on a Border Element, page 185
- Configuring SIP Ability to Send a SIP Registration Message on a Border Element, page 186
- Feature Information for Sending a SIP Registration Message from a Cisco Unified Border Element, page 187

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for SIP Ability to Send a SIP Registration Message on a Border Element

- Configure a registrar in sip UA configuration mode.

Cisco Unified Border Element

- Cisco IOS Release 12.4(24)T or a later release must be installed and running on your Cisco Unified Border Element.

Cisco Unified Border Element (Enterprise)
Cisco IOS XE Release 2.5 or a later release must be installed and running on your Cisco ASR 1000 Series Router.

Configuring SIP Ability to Send a SIP Registration Message on a Border Element

The SIP: Ability to Send a SIP Registration Message on a Border Element feature allows users to register e164 numbers from the Cisco UBE without POTS dial-peers in the UP state. Registration messages can include numbers, number ranges (such as E.164-numbers), or text information.

**SUMMARY STEPS**

1. enable
2. configure terminal
3. sip-ua
4. credentials username username password password realm domain-name
5. exit
6. end

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** enable | Enables privileged EXEC mode.  
• Enter your password if prompted. |
| Example: Device> enable |
| **Step 2** configure terminal | Enters global configuration mode. |
| Example: Device# configure terminal |
| **Step 3** sip-ua | Enters sip user-agent configuration mode. |
| Example: Device(config)# sip-ua |
| **Step 4** credentials username username password password realm domain-name | Enters SIP digest credentials in sip-ua configuration mode. |
| Example: Device(config-sip-ua)# credentials username alex password test realm cisco.com |
Purpose
Command or Action | Purpose
--- | ---
Step 5 | exit
Example: Device(config-sip-ua)# exit
Exitsthecurrentmode.
Step 6 | end
Example: Device(config)# end
Returns to privileged EXEC mode.

Feature Information for Sending a SIP Registration Message from a Cisco Unified Border Element

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 . An account on Cisco.com is not required.

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIP: Ability to Send a SIP Registration Message on a Border Element</td>
<td>12.4(24)T</td>
<td>Provides the ability to send a SIP Registration Message from Cisco Unified Border Element. The following command was modified: credentials (SIP UA)</td>
</tr>
<tr>
<td>SIP: Ability to Send a SIP Registration Message on a Border Element</td>
<td>Cisco IOS XE Release 2.5</td>
<td>Provides the ability to send a SIP Registration Message from Cisco Unified Border Element. The following command was modified: credentials (SIP UA)</td>
</tr>
</tbody>
</table>
Session Refresh with Reinvites

• Finding Feature Information, page 189
• Prerequisites for Session Refresh with Reinvites, page 189
• Information about Session Refresh with Reinvites, page 190
• How to Configure Session Refresh with Reinvites, page 190
• Feature Information for Session Refresh with Reinvites, page 192

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Session Refresh with Reinvites

The allow-connections sip to sip command must be configured before you configure the Session refresh with Reinvites feature. For more information and configuration steps see the "Configuring SIP-to-SIP Connections in a Cisco Unified Border Element" section.

Cisco Unified Border Element

• Cisco IOS Release 12.4(20)T or a later release must be installed and running on your Cisco Unified Border Element.

Cisco Unified Border Element (Enterprise)

• Cisco IOS XE Release 2.5 or a later release must be installed and running on your Cisco ASR 1000 Series Router.
Information about Session Refresh with Reinvites

Configuring support for session refresh with reinvites expands the ability of the Cisco Unified Border Element to receive a REINVITE message that contains either a session refresh parameter or a change in media via a new SDP and ensure the session does not time out. The midcall-signaling command distinguishes between the way a Cisco Unified Communications Express and Cisco Unified Border Element releases signaling messages. Most SIP-to-SIP video and SIP-to-SIP ReInvite-based supplementary services features require the Configuring Session Refresh with Reinvites feature to be configured.

Cisco IOS Release 12.4(15)XZ and Earlier Releases
Session refresh support via OPTIONS method. For configuration information, see the "Enabling In-Dialog OPTIONS to Monitor Active SIP Sessions" section.

Cisco IOS Release 12.4(15)XZ and Later Releases
Cisco Unified BE transparently passes other session refresh messages and parameters so that UAs and proxies can establish keepalives on a call.

How to Configure Session Refresh with Reinvites

Configuring Session refresh with Reinvites

Before You Begin

---

Note

SIP-to-SIP video calls and SIP-to-SIP ReInvite-based supplementary services fail if the midcall-signaling command is not configured.

---

Note

The following features function if the midcall-signaling command is not configured: session refresh, fax, and refer-based supplementary services.

- Configuring Session Refresh with Reinvites is for SIP-to-SIP calls only. All other calls (H323-to-SIP, and H323-to-H323) do not require the midcall-signaling command be configured
- Configuring the Session Refresh with Reinvites feature on a dial-peer basis is not supported.
SUMMARY STEPS

1. enable
2. configure terminal
3. voice service voip
4. sip
5. midcall-signaling passthru
6. exit
7. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| 1    | enable            | Enables privileged EXEC mode.  
|      |                   | • Enter your password if prompted. |
| 2    | configure terminal | Enters global configuration mode. |
| 3    | voice service voip | Enters VoIP voice-service configuration mode. |
| 4    | sip               | Enters SIP configuration mode. |
| 5    | midcall-signaling passthru | Passes SIP messages from one IP leg to another IP leg. |
| 6    | exit              | Exits the current mode. |
Feature Information for Session Refresh with Reinvites

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 . An account on Cisco.com is not required.

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session Refresh with Reinvites</td>
<td>12.4(20)T</td>
<td>Expands the ability of the Cisco Unified BE to control the session refresh parameters and ensure the session does not time out.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>In Cisco IOS Release 12.4(20)T, this feature was implemented on the Cisco Unified Border Element.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>midcall-signaling</strong></td>
</tr>
<tr>
<td>Session Refresh with Reinvites</td>
<td>Cisco IOS XE Release 2.5</td>
<td>Expands the ability of the Cisco Unified BE to control the session refresh parameters and ensure the session does not time out.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>In Cisco IOS XE Release 2.5, this feature was implemented on the Cisco Unified Border Element (Enterprise).</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>midcall-signaling</strong></td>
</tr>
</tbody>
</table>
CHAPTER 21

SIP Stack Portability

Implements capabilities to the SIP gateway Cisco IOS stack involving user-agent handling of messages, handling of unsolicited messages, support for outbound delayed media, and SIP headers and content in requests and responses.

- Finding Feature Information, page 193
- Prerequisites for SIP Stack Portability, page 193
- Information About SIP Stack Portability, page 194
- SIP Call-Transfer Basics, page 194
- Feature Information for SIP Stack Portability, page 205

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for SIP Stack Portability

Cisco Unified Border Element

- Cisco IOS Release 12.4(2)T or a later release must be installed and running on your Cisco Unified Border Element.

Cisco Unified Border Element (Enterprise)

- Cisco IOS XE Release 2.5 or a later release must be installed and running on your Cisco ASR 1000 Series Router.
Information About SIP Stack Portability

The SIP Stack Portability feature implements the following capabilities to the Cisco IOS SIP gateway stack:

- It receives inbound Refer message requests both within a dialog and outside of an existing dialog from the user agents (UAs).
- It sends and receives SUBSCRIBE or NOTIFY message requests via UAs.
- It receives unsolicited NOTIFY message requests without having to subscribe to the event that was generated by the NOTIFY message request.
- It supports outbound delayed media.

It sends an INVITE message request without Session Description Protocol (SDP) and provides SDP information in either the PRACK or ACK message request for both initial call establishment and mid-call re-INVITE message requests.

- It sets SIP headers and content body in requests and responses.

The stack applies certain rules and restrictions for a subset of headers and for some content types (such as SDP) to protect the integrity of the stack’s functionality and to maintain backward compatibility. When receiving SIP message requests, it reads the SIP header and any attached body without any restrictions.

To make the best use of SIP call-transfer features, you should understand the following concepts:

SIP Call-Transfer Basics

Basic Terminology of SIP Call Transfer

Call transfer allows a wide variety of decentralized multiparty call operations. These decentralized call operations form the basis for third-party call control, and thus are important features for VoIP and SIP. Call transfer is also critical for conference calling, where calls can transition smoothly between multiple point-to-point links and IP-level multicasting.

Refer Message Request

The SIP Refer message request provides call-transfer capabilities to supplement the SIP BYE and ALSO message requests already implemented on Cisco IOS SIP gateways. The Refer message request has three main roles:

- Originator--User agent that initiates the transfer or Refer request.
- Recipient--User agent that receives the Refer request and is transferred to the final-recipient.
- Final-Recipient--User agent introduced into a call with the recipient.

Note: A gateway can be a recipient or final recipient, but not an originator.
The Refer message request always begins within the context of an existing call and starts with the *originator*. The originator sends a Refer request to the *recipient* (user agent receiving the Refer request) to initiate a triggered INVITE request. The triggered INVITE request uses the SIP URL contained in the Refer-To header as the destination of the INVITE request. The recipient then contacts the resource in the Refer-To header (*final recipient*), and returns a SIP 202 (Accepted) response to the originator. The recipient also must notify the originator of the outcome of the Refer transaction—whether the final recipient was successfully contacted or not. The notification is accomplished using the SIP NOTIFY message request, SIP’s event notification mechanism. A NOTIFY message with a message body of SIP 200 OK indicates a successful transfer, and a message body of SIP 503 Service Unavailable indicates an unsuccessful transfer. If the call was successful, a call between the recipient and the final recipient results.

The figure below represents the call flow of a successful Refer transaction initiated within the context of an existing call.

**Figure 6: Successful Refer transaction**

```
<table>
<thead>
<tr>
<th>Originator</th>
<th>Recipient</th>
<th>Final recipient</th>
</tr>
</thead>
<tbody>
<tr>
<td>user agent A</td>
<td>user agent B</td>
<td>user agent C</td>
</tr>
</tbody>
</table>

INVITE/200/ACK

2-Way RTP

Refer: Refer-To: Agent C

202 Accepted

Notify (100 Trying body)

200 OK

INVITE

100 Trying

200 OK

Notify 200 OK (Refer success)

200 OK
```

**Refer-To Header**

The recipient receives from the originator a Refer request that always contains a single Refer-To header. The Refer-To header includes a SIP URL that indicates the party to be invited and must be in SIP URL format.

*Note*

The TEL URL format cannot be used in a Refer-To header, because it does not provide a host portion, and without one, the triggered INVITE request cannot be routed.

The Refer-To header may contain three additional overloaded headers to form the triggered INVITE request. If any of these three headers are present, they are included in the triggered INVITE request. The three headers are:
Basic Terminology of SIP Call Transfer

- Accept-Contact--Optional in a Refer request. A SIP Cisco IOS gateway that receives an INVITE request with an Accept-Contact does not act upon this header. This header is defined in draft-ietf-sip-callerprefs-03.txt and may be used by user agents that support caller preferences.

- Proxy-Authorization--Nonstandard header that SIP gateways do not act on. It is echoed in the triggered INVITE request because proxies occasionally require it for billing purposes.

- Replaces--Header used by SIP gateways to indicate whether the originator of the Refer request is requesting a blind or attended transfer. It is required if the originator is performing an attended transfer, and not required for a blind transfer.

All other headers present in the Refer-To are ignored, and are not sent in the triggered INVITE.

Note

The Refer-To and Contact headers are required in the Refer request. The absence of these headers results in a 4xx class response to the Refer request. Also, the Refer request must contain exactly one Refer-To header. Multiple Refer-To headers result in a 4xx class response.

Referred-By Header

The Referred-By header is required in a Refer request. It identifies the originator and may also contain a signature (included for security purposes). SIP gateways echo the contents of the Referred-By header in the triggered INVITE request, but on receiving an INVITE request with this header, gateways do not act on it.

Note

The Referred-By header is required in a Refer request. The absence of this header results in a 4xx class response to the Refer request. Also, the Refer request must contain exactly one Referred-By header. Multiple Referred-By headers result in a 4xx class response.

NOTIFY Message Request

Once the outcome of the Refer transaction is known, the recipient of the Refer request must notify the originator of the outcome of the Refer transaction--whether the final-recipient was successfully contacted or not. The notification is accomplished using the NOTIFY message request, SIP’s event notification mechanism. The notification contains a message body with a SIP response status line and the response class in the status line indicates the success or failure of the Refer transaction.

The NOTIFY message must do the following:

- Reflect the same To, From, and Call-ID headers that were received in the Refer request.
- Contain an Event header refer.
- Contain a message body with a SIP response line. For example: SIP/2.0 200 OK to report a successful Refer transaction, or SIP/2.0 503 Service Unavailable to report a failure. To report that the recipient disconnected before the transfer finished, it must use SIP/2.0 487 Request Canceled.

Two Cisco IOS commands pertain to the NOTIFY message request:

- The timers notify command sets the amount of time that the recipient should wait before retransmitting a NOTIFY message to the originator.
• The `retry notify` command configures the number of times a NOTIFY message is retransmitted to the originator.

**Note**
For information on these commands, see the *Cisco IOS Voice Command Reference*.

---

**Types of SIP Call Transfer Using the Refer Message Request**

This section discusses how the Refer message request facilitates call transfer.

There are two types of call transfer: blind and attended. The primary difference between the two is that the Replaces header is used in attended call transfers. The Replaces header is interpreted by the final recipient and contains a Call-ID header, indicating that the initial call leg is to be replaced with the incoming INVITE request.

As outlined in the Refer message request, there are three main roles:

- **Originator**--User agent that initiates the transfer or Refer request.
- **Recipient**--User agent that receives the Refer request and is transferred to the final recipient.
- **Final-Recipient**--User agent introduced into a call with the recipient.

A gateway can be a recipient or final recipient, but not an originator.

**Blind Call-Transfer Process**

A blind, or unattended, transfer is one in which the transferring phone connects the caller to a destination line before ringback begins. This is different from a consultative, or attended, transfer in which one of the transferring parties either connects the caller to a ringing phone (ringback heard) or speaks with the third party before connecting the caller to the third party. Blind transfers are often preferred by automated devices that do not have the capability to make consultation calls.

Blind transfer works as described in the *Types of SIP Call Transfer Using the Refer Message Request*, on page 197. The process is as follows:

1. **Originator** (user agent that initiates the transfer or Refer request) does the following:
   1. Sets up a call with **recipient** (user agent that receives the Refer request)
   2. Issues a Refer request to **recipient**

2. **Recipient** does the following:
   1. Sends an INVITE request to **final recipient** (user agent introduced into a call with the recipient)
   2. Returns a SIP 202 (Accepted) response to originator
   3. Notifies originator of the outcome of the Refer transaction—whether final recipient was successfully (SIP 200 OK) contacted or not (SIP 503 Service Unavailable)

3. If successful, a call is established between **recipient** and **final recipient**.

4. The original signaling relationship between originator and recipient terminates when either of the following occurs:

5. One of the parties sends a Bye request.
Recipient sends a Bye request after successful transfer (if originator does not first send a Bye request after receiving an acknowledgment for the NOTIFY message).

The figure below shows a successful blind or unattended call transfer in which the originator initiates a Bye request to terminate signaling with the recipient.

**Figure 7: Successful Blind or Unattended Transfer--Originator Initiating a Bye Request**

<table>
<thead>
<tr>
<th>Originator</th>
<th>Recipient</th>
<th>Final recipient</th>
</tr>
</thead>
<tbody>
<tr>
<td>INVITE/200OK/ACK</td>
<td></td>
<td>INVITE</td>
</tr>
<tr>
<td>2-way RTP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>REFER (refer-to Final recipient)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>202 Accepted</td>
<td></td>
<td>100 Trying</td>
</tr>
<tr>
<td>Notify (100 Trying body)</td>
<td></td>
<td>INVITE (referred-by recipient)</td>
</tr>
<tr>
<td>200 OK</td>
<td></td>
<td>18x/200</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>200 OK/ACK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2-way RTP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>200 OK BYE</td>
</tr>
</tbody>
</table>
The figure below shows a successful blind or unattended call transfer in which the recipient initiates a Bye request to terminate signaling with the originator. A NOTIFY message is always sent by the recipient to the originator after the final outcome of the call is known.

*Figure 8: Successful Blind or Unattended Transfer--Recipient Initiating a Bye Request*

If a failure occurs with the triggered INVITE to the final recipient, the call between originator and recipient is not disconnected. Rather, with blind transfer the process is as follows:

1. Originator sends a re-INVITE that takes the call off hold and returns to the original call with recipient.
2. Final recipient sends an 18x informational response to recipient.
3. The call fails; the originator cannot recover the call with recipient. Failure can be caused by an error condition or timeout.
4. The call leg between originator and recipient remains active (see the figure below).
5. If the INVITE to final recipient fails (408 Request Timeout), the following occurs:
   1. Recipient notifies originator of the failure with a NOTIFY message.
2 Originator sends a re-INVITE and returns to the original call with the recipient.

**Figure 9: Failed Blind Transfer—Originator Returns to Original Call with Recipient**

---

**Attended Transfer**

In attended transfers, the Replaces header is inserted by the initiator of the Refer message request as an overloaded header in the Refer-To and is copied into the triggered INVITE request sent to the final recipient. The header has no effect on the recipient, but is interpreted by the final recipient as a way to distinguish between blind transfer and attended transfer. The attended transfer process is as follows:

1 Originator does the following:
   1 Sets up a call with recipient.
   2 Places recipient on hold.
   3 Establishes a call to final recipient.
   4 Sends recipient a Refer message request with an overloaded Replaces header in the Refer-To header.

2 Recipient does the following:
   1 Sends a triggered INVITE request to final recipient. (Request includes the Replaces header, identifying the call leg between the originator and the final recipient.)
2 Recipient returns a SIP 202 (Accepted) response to originator. (Response acknowledges that the INVITE has been sent.)

3 Final recipient establishes a direct signaling relationship with recipient. (Replaces header indicates that the initial call leg is to be shut down and replaced by the incoming INVITE request.)

4 Recipient notifies originator of the outcome of the Refer transaction. (Outcome indicates whether or not the final recipient was successfully contacted.)

5 Recipient terminates the session with originator by sending a Bye request.

Replaces Header

The Replaces header is required in attended transfers. It indicates to the final recipient that the initial call leg (identified by the Call-ID header and tags) is to be shut down and replaced by the incoming INVITE request. The final recipient sends a Bye request to the originator to terminate its session.

If the information provided by the Replaces header does not match an existing call leg, or if the information provided by the Replaces header matches a call leg but the call leg is not active (a Connect, 200 OK to the INVITE request has not been sent by the final-recipient), the triggered INVITE does not replace the initial call leg and the triggered INVITE request is processed normally.

Any failure resulting from the triggered INVITE request from the recipient to the final recipient does not drop the call between the originator and the final recipient. In these scenarios, all calls that are active (originator to recipient and originator to final recipient) remain active after the failed attended transfer attempt.
The figure below shows a call flow for a successful attended transfer.

**Figure 10: Successful Attended Transfer**

**Attended Transfer with Early Completion**

Attended transfers allow the originator to have a call established between both the recipient and the final recipient. With attended transfer with early completion, the call between the originator and the final recipient does not have to be active, or in the talking state, before the originator can transfer it to the recipient. The
originator establishes a call with the recipient and only needs to be setting up a call with the final recipient. The final recipient may be ringing, but has not answered the call from the originator when it receives a re-INVITE to replace the call with the originator and the recipient.

The process for attended transfer with early completion is as follows (see the figure below):

1. Originator does the following:
   1. Sets up a call with recipient.
   2. Places the recipient on hold.
   3. Contacts the final recipient.
   4. After receiving an indication that the final recipient is ringing, sends recipient a Refer message request with an overloaded Replaces header in the Refer-To header. (The Replaces header is required in attended transfers and distinguishes between blind transfer and attended transfers.)

2. Recipient does the following:
   1. Returns a SIP 202 (Accepted) response to the originator. (to acknowledge that the INVITE has been sent.)
   2. Upon receipt of the Refer message request, sends a triggered INVITE request to final recipient. (The request includes the Replaces header, which indicates that the initial call leg, as identified by the Call-ID header and tags, is to be shut down and replaced by the incoming INVITE request.)
   3. Final recipient establishes a direct signaling relationship with recipient.
   4. Final recipient tries to match the Call-ID header and the To or From tag in the Replaces header of the incoming INVITE with an active call leg in its call control block. If a matching active call leg is found, final recipient replies with the same status as the found call leg. However, it then terminates the found call leg with a 487 Request Cancelled response.

Note

If early transfer is attempted and the call involves quality of service (QoS) or Resource Reservation Protocol (RSVP), the triggered INVITE from the recipient with the Replaces header is not processed and the transfer fails. The session between originator and final recipient remains unchanged.

1. Recipient notifies originator of the outcome of the Refer transaction--that is, whether final recipient was successfully contacted or not.
Recipient or originator terminates the session by sending a Bye request.

**Figure 11: Attended Transfer with Early Completion**

<table>
<thead>
<tr>
<th>Originator</th>
<th>Recipient</th>
<th>Final recipient</th>
</tr>
</thead>
<tbody>
<tr>
<td>INVITE/200/ACK&lt;br&gt;Call-id:1;from-tag:11;to-tag:22</td>
<td>2-way RTP&lt;br&gt;INVITE(hold)/200/ACK&lt;br&gt;Call-id:1;from-tag:11;to-tag:22</td>
<td>RTP on hold&lt;br&gt;18x Call-id:2;from-tag:33;to-tag:44</td>
</tr>
<tr>
<td>Complete transfer early&lt;br&gt;Refer (Refer-To: final-recipient?) Replaces Call-id:2;from-tag:33;to-tag:44&lt;br&gt;Call-id:1;from-tag:to-tag:22</td>
<td>SIP 202 Accepted&lt;br&gt;Notify (100 Trying body)</td>
<td>Invite&lt;br&gt;100 Trying&lt;br&gt;18x Call-id:3;from-tag:55;to-tag:66&lt;br&gt;Replaces: Call-id:2;from-tag:33;to-tag:44</td>
</tr>
<tr>
<td>200 OK</td>
<td>Invite</td>
<td>487 Request Cancelled Call-id:2;from-tag:33;to-tag:44&lt;br&gt;ACK</td>
</tr>
<tr>
<td>2-Way RTP</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**VSA for Call Transfer**

You can use a vendor-specific attribute (VSA) for SIP call transfer.

**Referred-By Header**

For consistency with existing billing models, Referred-By and Requested-By headers are populated in call history tables as a VSA. Cisco VSAs are used for VoIP call authorization. The new VSA tag
supp-svc-xfer-by helps to associate the call legs for call-detail-record (CDR) generation. The call legs can be originator-to-recipient or recipient-to-final-recipient.

The VSA tag supp-svc-xfer-by contains the user@host portion of the SIP URL of the Referred-By header for transfers performed with the Refer message request. For transfers performed with the Bye/Also message request, the tag contains user@host portion of the SIP URL of the Requested-By header. For each call on the gateway, two RADIUS records are generated: start and stop. The supp-svc-xfer-by VSA is generated only for stop records and is generated only on the recipient gateway—the gateway receiving the Refer or Bye/Also message.

The VSA is generated when a gateway that acts as a recipient receives a Refer or Bye/Also message with the Referred-By or Requested-By headers. There are usually two pairs of start and stop records. There is a start and stop record between the recipient and the originator and also between the recipient to final recipient. In the latter case, the VSA is generated between the recipient to the final recipient only.

**Business Group Field**

A new business group VSA field has been added that assists service providers with billing. The field allows service providers to add a proprietary header to call records. The VSA tag for business group ID is cust-biz-grp-id and is generated only for stop records. It is generated when the gateway receives an initial INVITE with a vendor dial-plan header to be used in call records. In cases when the gateway acts as a recipient, the VSA is populated in the stop records between the recipient and originator and the final recipient.

Note

For information on VSAs, see the RADIUS VSA Voice Implementation Guide.

---

**Feature Information for SIP Stack Portability**

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 . An account on Cisco.com is not required.

**Table 20: Feature Information for SIP Stack Portability**

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIP Stack Portability</td>
<td>Cisco IOS XE Release 2.5</td>
<td>Implements capabilities to the SIP gateway Cisco IOS stack involving user-agent handling of messages, handling of unsolicited messages, support for outbound delayed media, and SIP headers and content in requests and responses. The following commands were introduced or modified: <strong>None</strong></td>
</tr>
</tbody>
</table>

---
<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIP Stack Portability</td>
<td>12.4(2)T</td>
<td>Implements capabilities to the SIP gateway Cisco IOS stack involving user-agent handling of messages, handling of unsolicited messages, support for outbound delayed media, and SIP headers and content in requests and responses. The following commands were introduced or modified: <strong>None</strong></td>
</tr>
</tbody>
</table>
Interworking of Secure RTP calls for SIP and H.323

The Session Initiation Protocol (SIP) support for the Secure Real-time Transport Protocol (SRTP) is an extension of the Real-time Transport Protocol (RTP) Audio/Video Profile (AVP) and ensures the integrity of RTP and Real-Time Control Protocol (RTCP) packets that provide authentication, encryption, and the integrity of media packets between SIP endpoints.

SIP support for SRTP was introduced in Cisco IOS Release 12.4(15)T. In this and later releases, you can configure the handling of secure RTP calls on both a global level and on an individual dial peer basis on Cisco IOS voice gateways. You can also configure the gateway (or dial peer) either to fall back to (nonsecure) RTP or to reject (fail) the call for cases where an endpoint does not support SRTP.

The option to allow negotiation between SRTP and RTP endpoints was added for Cisco IOS Release 12.4(20)T and later releases, as was interoperability of SIP support for SRTP on Cisco IOS voice gateways with Cisco Unified Communications Manager. In Cisco IOS Release 12.4(22)T and later releases, you can also configure SIP support for SRTP on Cisco Unified Border Elements (Cisco UBEs).

Finding Feature Information, page 207
Prerequisites for Interworking of Secure RTP calls for SIP and H.323, page 208
Restrictions for Interworking of Secure RTP calls for SIP and H.323, page 208
Feature Information for Configuring Interworking of Secure RTP Calls for SIP and H.323, page 209

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.
Prerequisites for Interworking of Secure RTP calls for SIP and H.323

The following are prerequisites for the Interworking of Secure RTP calls for SIP and H.323 feature:

• Establish a working IP network and configure VoIP.

For information about configuring VoIP, see Enhancements to the Session Initiation Protocol for VoIP on Cisco Access Platforms at the following URL:

• Ensure that the gateway has voice functionality configured for SIP.

• Ensure that your Cisco router has adequate memory.

• As necessary, configure the router to use Greenwich Mean Time (GMT). SIP requires that all times be sent in GMT. SIP INVITE messages are sent in GMT. However, the default for routers is to use Coordinated Universal Time (UTC). To configure the router to use GMT, issue the clock timezone command in global configuration mode and specify GMT.

Cisco Unified Border Element

• Cisco IOS Release 12.2(20)T 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.

Restrictions for Interworking of Secure RTP calls for SIP and H.323

• The SIP gateway does not support codecs other than those listed in the table titled "SIP Codec Support by Platform and Cisco IOS Release" in the "Enhanced Codec Support for SIP Using Dynamic Payloads" section of the Configuring SIP QoS Features module at the following URL:

• SIP requires that all times be sent in GMT.
Feature Information for Configuring Interworking of Secure RTP Calls for SIP and H.323

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 . An account on Cisco.com is not required.

Table 21: Feature Information for Configuring Support for Expires Timer Reset on Receiving or Sending SIP 183 Message

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interworking of Secure RTP calls for SIP and H.323</td>
<td>12.4(20)T</td>
<td>This feature provides an option for a Secure RTP (SRTP) call to be connected from H.323 to SIP and from SIP to H.323. Additionally, this feature extends SRTP fallback support from the Cisco IOS voice gateway to the Cisco Unified Border Element. This feature uses no new or modified commands.</td>
</tr>
<tr>
<td>Interworking of Secure RTP calls for SIP and H.323</td>
<td>Cisco IOS XE Release 3.1S</td>
<td>This feature provides an option for a Secure RTP (SRTP) call to be connected from H.323 to SIP and from SIP to SIP. Additionally, this feature extends SRTP fallback support from the Cisco IOS voice gateway to the Cisco Unified Border Element. This feature uses no new or modified commands.</td>
</tr>
</tbody>
</table>
Cisco UBE Support for SRTP-RTP Internetworking

The Cisco Unified Border Element Support for SRTP-RTP Internetworking feature allows secure enterprise-to-enterprise calls and provides operational enhancements for Session Initiation Protocol (SIP) trunks from Cisco Unified Call Manager and Cisco Unified Call Manager Express. Support for Secure Real-Time Transport Protocol (SRTP)-Real-Time Transport Protocol (RTP) internetworking between one or multiple Cisco Unified Border Elements (Cisco UBEs) is enabled for SIP-SIP audio calls.

In Cisco IOS Release 15.2(1) and Cisco IOS XE Release 3.7S, the SRTP-RTP Interworking feature was extended to support supplementary services on Cisco UBEs.

- Prerequisites for CUBE Support for SRTP-RTP Internetworking, page 211
- Restrictions for CUBE Support for SRTP-RTP Internetworking, page 212
- Information About CUBE for SRTP-RTP Internetworking, page 212
- How to Configure Cisco UBE Support for SRTP-RTP Internetworking, page 215
- Configuration Examples for CUBE Support for SRTP-RTP Internetworking, page 233
- Feature Information for CUBE Support for SRTP-RTP Internetworking, page 235

Prerequisites for CUBE Support for SRTP-RTP Internetworking

- The Cisco Unified Border Element Support for SRTP-RTP Internetworking feature is supported in Cisco Unified CallManager 7.0 and later releases.

Cisco Unified Border Element

- Cisco IOS Release 12.4(22)YB or a later release must be installed and running on your Cisco Unified Border Element.

Cisco Unified Border Element (Enterprise)

- Cisco IOS XE Release 3.7S or a later release must be installed and running on your Cisco ASR 1000 Series Router.
Restrictions for CUBE Support for SRTP-RTP Internetworking

The following features are not supported by the Cisco Unified Border Element Support for SRTP-RTP Internetworking feature:

- Asymmetric SRTP fallback configurations
- Call admission control (CAC) support
- Rotary SIP-SIP
- SRTCP-RTCP interworking
- Transcoding for SRTP-SRTP audio calls

Note: Effective from Cisco IOS XE release 3.9S, SRTP-RTP interworking is supported (on ASR platforms) for video calls with no secondary video streams.

Information About CUBE for SRTP-RTP Internetworking

To configure support for SRTP-RTP internetworking, you should understand the following concepts:

CUBE Support for SRTP-RTP Internetworking

The Cisco Unified Border Element Support for SRTP-RTP Internetworking feature connects SRTP Cisco Unified CallManager domains with the following:

- RTP Cisco Unified CallManager domains. Domains that do not support SRTP or have not been configured for SRTP, as shown in the figure below.

- RTP Cisco applications or servers. For example, Cisco Unified MeetingPlace, Cisco WebEx, or Cisco Unity, which do not support SRTP, or have not been configured for SRTP, or are resident in a secure data center, as shown in the figure below.
• RTP to third-party equipment. For example, IP trunks to PBXs or virtual machines, which do not support SRTP.

**Figure 12: SRTP Domain Connections**

The Cisco Unified Border Element Support for SRTP-RTP Internetworking feature connects SRTP enterprise domains to RTP SIP provider SIP trunks. SRTP-RTP internetworking connects RTP enterprise networks with SRTP over an external network between businesses. This provides flexible secure business-to-business communications without the need for static IPsec tunnels or the need to deploy SRTP within the enterprise, as shown in the figure below.

**Figure 13: Secure Business-to-Business Communications**
SRTP-RTP internetworking also connects SRTP enterprise networks with static IPsec over external networks, as shown in the figure below.

*Figure 14: SRTP Enterprise Network Connections*

SRTP-RTP internetworking on the Cisco UBE in a network topology uses single-pair key generation. Existing audio and dual-tone multifrequency (DTMF) transcoding is used to support voice calls. SRTP-RTP internetworking support is provided in both flow-through and high-density mode. SRTP-SRTP pass-through is not impacted.

SRTP is configured on one dial peer and RTP is configured on the other dial peer using the `srtp` and `srtp fallback` commands. The dial-peer configuration takes precedence over the global configuration on the Cisco UBE.

Fallback handling occurs if one of the call endpoints does not support SRTP. The call can fall back to RTP-RTP, or the call can fail, depending on the configuration. Fallback takes place only if the `srtp fallback` command is configured on the respective dial peer. RTP-RTP fallback occurs when no transcoding resources are available for SRTP-RTP internetworking.

**TLS on the Cisco Unified Border Element**

The Cisco Unified Border Element Support for SRTP-RTP Internetworking feature allows Transport Layer Security (TLS) to be enabled or disabled between the Skinny Call Control Protocol (SCCP) server and the SCCP client. By default, TLS is enabled, which provides added protection at the transport level and ensures that SRTP keys are not easily accessible. Once TLS is disabled, the SRTP keys are not protected.

SRTP-RTP internetworking is available with normal and universal transcoders. The transcoder on the Cisco Unified Border Element is invoked using SCCP messaging between the SCCP server and the SCCP client. SCCP messages carry the SRTP keys to the digital signal processor (DSP) farm at the SCCP client. The transcoder can be within the same router or can be located in a separate router. TLS should be disabled only when the transcoder is located in the same router. To disable TLS, configure the `no` form of the `tls` command in `dsp farm profile configuration mode`. Disabling TLS improves CPU performance.

**Supplementary Services Support on the Cisco UBE for RTP-SRTP Calls**

The Supplementary Services Support on Cisco UBE for RTP-SRTP Calls feature supports the following supplementary services on the Cisco UBE:

- Midcall codec change with voice class codec configuration for SRTP-RTP and SRTP pass-through calls.
• Reinvite-based call hold.
• Reinvite-based call resume.
• Music on hold (MoH) invoked from the Cisco Unified Communications Manager (Cisco UCM), where the call leg changes between SRTP and RTP for an MoH source.
  Reinvite-based call forward.
• Reinvite-based call transfer.
• Call transfer based on a REFER message, with local consumption or pass-through of the REFER message on the Cisco UBE.
• Call forward based on a 302 message, with local consumption or pass-through of the 302 message on the Cisco UBE.
• T.38 fax switchover.
• Fax pass-through switchover.
• DO-EO for SRTP-RTP calls.
• DO-EO for SRTP pass-through calls.

When the initial SRTP-RTP or SRTP pass-through call is established on the Cisco UBE, a call can switch between SRTP and RTP for various supplementary services that can be invoked on the end points. Transcoder resources are used to perform SRTP-RTP conversion on Cisco UBE. When the call switches between SRTP and RTP, the transcoder is dynamically inserted, deleted, or modified. Both normal transcoding and high-density (optimized) transcoding are supported.

For call transfers involving REFER and 302 messages (messages that are locally consumed on Cisco UBE), end-to-end media renegotiation is initiated from Cisco UBE only when you configure the supplementary-service media-renegotiate command in voice service voice configuration mode.

When supplementary services are invoked from the end points, the call can switch between SRTP and RTP during the call duration. Hence, Cisco recommends that you configure such SIP trunks for SRTP fallback.

How to Configure Cisco UBE Support for SRTP-RTP Internetworking

Configuring Cisco UBE Support for SRTP-RTP Internetworking

Configuring the Certificate Authority

Perform the steps described in this section to configure the certificate authority.
SUMMARY STEPS

1. enable
2. configure terminal
3. ip http server
4. crypto pki server cs-label
5. database level complete
6. grant auto
7. no shutdown
8. exit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td></td>
</tr>
<tr>
<td>enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device&gt; enable</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td></td>
</tr>
<tr>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td></td>
</tr>
<tr>
<td>ip http server</td>
<td>Enables the HTTP server on your IPv4 or IPv6 system, including the Cisco web browser user interface.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device(config)# ip http server</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td></td>
</tr>
<tr>
<td>crypto pki server cs-label</td>
<td>Enables a Cisco IOS certificate server and enters certificate server configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>• In the example, 3854-cube is specified as the name of the certificate server.</td>
</tr>
<tr>
<td>Device(config)# crypto pki server 3854-cube</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td></td>
</tr>
<tr>
<td>database level complete</td>
<td>Controls what type of data is stored in the certificate enrollment database.</td>
</tr>
<tr>
<td>Example:</td>
<td>• In the example, each issued certificate is written to the database.</td>
</tr>
<tr>
<td>Device(cs-server)# database level complete</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>grant auto</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(cs-server)# grant auto</td>
</tr>
<tr>
<td></td>
<td>Specifies automatic certificate enrollment.</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>no shutdown</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(cs-server)# no shutdown</td>
</tr>
<tr>
<td></td>
<td>Reenables the certificate server.</td>
</tr>
<tr>
<td></td>
<td>• Create and enter a new password when prompted.</td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>exit</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(cs-server)# exit</td>
</tr>
<tr>
<td></td>
<td>Exits certificate server configuration mode.</td>
</tr>
</tbody>
</table>

### Configuring a Trustpoint for the Secure Universal Transcoder

Perform the task in this section to configure, authenticate, and enroll a trustpoint for the secure universal transcoder.

**Before You Begin**

Before you configure a trustpoint for the secure universal transcoder, you should configure the certificate authority, as described in the Configuring the Certificate Authority, on page 215.

**SUMMARY STEPS**

1. enable
2. configure terminal
3. crypto pki trustpoint name
4. enrollment url url
5. serial-number
6. revocation-check method
7. rsakeypair key-label
8. end
9. crypto pki authenticate name
10. crypto pki enroll name
11. exit
## Detailed Steps

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| Step 1 | `enable` | Enables privileged EXEC mode.  
- Enter your password if prompted.  
  | `Example:`  
  Device> enable |
| Step 2 | `configure terminal` | Enters global configuration mode.  
  | `Example:`  
  Device# configure terminal |
| Step 3 | `crypto pki trustpoint name` | Declares the trustpoint that the router uses and enters ca-trustpoint configuration mode.  
- In the example, the trustpoint is named secdsp.  
  | `Example:`  
  Device(config)# crypto pki trustpoint secdsp |
| Step 4 | `enrollment url url` | Specifies the enrollment parameters of a certification authority (CA).  
- In the example, the URL is defined as http://10.13.2.52:80.  
  | `Example:`  
  Device(ca-trustpoint)# enrollment url http://10.13.2.52:80 |
| Step 5 | `serial-number` | Specifies whether the router serial number should be included in the certificate request.  
  | `Example:`  
  Device(ca-trustpoint)# serial-number |
| Step 6 | `revocation-check method` | Checks the revocation status of a certificate.  
- In the example, the certificate revocation list checks the revocation status.  
  | `Example:`  
  Device(ca-trustpoint)# revocation-check crl |
| Step 7 | `rsakeypair key-label` | Specifies which key pair to associate with the certificate.  
- In the example, the key pair 3845-cube generated during enrollment is associated with the certificate.  
  | `Example:`  
  Device(ca-trustpoint)# rsakeypair 3845-cube |
| Step 8 | `end` | Exits ca-trustpoint configuration mode.  
  | `Example:`  
  Device(ca-trustpoint)# end |
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 9** crypto pki authenticate name | Authenticates the CA.  
- Accept the trustpoint CA certificate if prompted. |
| Example: Device(config)# crypto pki authenticate secdsp | |
| **Step 10** crypto pki enroll name | Obtains the certificate for the router from the CA.  
- Create and enter a new password if prompted.  
- Request a certificate from the CA if prompted. |
| Example: Device(config)# crypto pki enroll secdsp | |
| **Step 11** exit | Exits global configuration mode. |
| Example: Device(config)# exit | |

### Configuring DSP Farm Services

Perform the task in this section to configure DSP farm services.

**Before You Begin**

Before you configure DSP farm services, you should configure the trustpoint for the secure universal transcoder, as described in the Configuring a Trustpoint for the Secure Universal Transcoder, on page 217.

**SUMMARY STEPS**

1. enable
2. configure terminal
3. voice-card slot
4. dspfarm
5. dsp services dspfarm
6. Repeat Steps 3, 4, and 5 to configure a second voice card.
7. exit

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
</tbody>
</table>
### Command or Action

**Step 2**

**configure terminal**

Example:

```
Device# configure terminal
```

**Step 3**

**voice-card slot**

Example:

```
Device(config)# voice-card 0
```

**Step 4**

**dspfarm**

Example:

```
Device(config-voicecard)# dspfarm
```

**Step 5**

**dsp services dspfarm**

Example:

```
Device(config-voicecard)# dsp services dspfarm
```

**Step 6**

Repeat Steps 3, 4, and 5 to configure a second voice card.

**Step 7**

**exit**

Example:

```
Device(config-voicecard)# exit
```

### Associating SCCP to the Secure DSP Farm Profile

Perform the task in this section to associate SCCP to the secure DSP farm profile.

**Before You Begin**

Before you associate SCCP to the secure DSP farm profile, you should configure DSP farm services, as described in the Configuring DSP Farm Services, on page 219.
SUMMARY STEPS

1. enable
2. configure terminal
3. sccp local interface-type interface-number
4. sccp ccm ip-address identifier identifier-number version version-number
5. sccp
6. associate ccm identifier-number priority priority-number
7. associate profile profile-identifier register device-name
8. dspfarm profile profile-identifier transcode universal security
9. trustpoint trustpoint-label
10. codec codec-type
11. Repeat Step 10 to configure required codecs.
12. maximum sessions number
13. associate application sccp
14. no shutdown
15. exit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>enable</td>
</tr>
<tr>
<td>Example:</td>
<td>Device&gt; enable</td>
</tr>
<tr>
<td></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>configure terminal</td>
</tr>
<tr>
<td>Example:</td>
<td>Device# configure terminal</td>
</tr>
<tr>
<td></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>sccp local interface-type interface-number</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config)# sccp local GigabitEthernet 0/0</td>
</tr>
<tr>
<td></td>
<td>Selects the local interface that SCCP applications (transcoding and conferencing) use to register with Cisco CallManager.</td>
</tr>
<tr>
<td></td>
<td>• In the example, the following parameters are set:</td>
</tr>
<tr>
<td></td>
<td>• GigabitEthernet is defined as the interface type that the SCCP application uses to register with Cisco CallManager.</td>
</tr>
<tr>
<td></td>
<td>• The interface number that the SCCP application uses to register with Cisco CallManager is specified as 0/0.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>sccp ccm ip-address identifier identifier-number version version-number</td>
</tr>
<tr>
<td></td>
<td>Adds a Cisco Unified Communications Manager server to the list of available servers.</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>In the example, the following parameters are set:</td>
</tr>
<tr>
<td><code>Device(config)# sccp ccm 10.13.2.52</code></td>
<td>• 10.13.2.52 is configured as the IP address of the Cisco Unified Communications Manager server.</td>
</tr>
<tr>
<td><code>identifier 1 version 5.0.1</code></td>
<td>• The number 1 identifies the Cisco Unified Communications Manager server.</td>
</tr>
<tr>
<td></td>
<td>• The Cisco Unified Communications Manager version is identified as 5.0.1.</td>
</tr>
</tbody>
</table>

**Step 5**

<table>
<thead>
<tr>
<th>sccp</th>
<th>Enables SCCP and related applications (transcoding and conferencing) and enters SCCP Cisco CallManager configuration mode.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Device(config)# sccp</code></td>
<td></td>
</tr>
</tbody>
</table>

**Step 6**

<table>
<thead>
<tr>
<th>associate <strong>ccm</strong> <code>identifier-number priority priority-number</code></th>
<th>Associates a Cisco Unified CallManager with a Cisco CallManager group and establishes its priority within the group.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>In the example, the following parameters are set:</td>
</tr>
<tr>
<td><code>Device(config-sccp-ccm)# associate ccm 1 priority 1</code></td>
<td>• The number 1 identifies the Cisco Unified CallManager.</td>
</tr>
<tr>
<td></td>
<td>• The Cisco Unified CallManager is configured with the highest priority within the Cisco CallManager group.</td>
</tr>
</tbody>
</table>

**Step 7**

<table>
<thead>
<tr>
<th>associate <strong>profile</strong> <code>profile-identifier register device-name</code></th>
<th>Associates a DSP farm profile with a Cisco CallManager group.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>In the example, the following parameters are set:</td>
</tr>
<tr>
<td><code>Device(config-sccp-ccm)# associate profile 1 register sxcoder</code></td>
<td>• The number 1 identifies the DSP farm profile.</td>
</tr>
<tr>
<td></td>
<td>• Sxcoder is configured as the user-specified device name in Cisco Unified CallManager.</td>
</tr>
</tbody>
</table>

**Step 8**

<table>
<thead>
<tr>
<th>dspfarm <strong>profile</strong> <code>profile-identifier transcode universal security</code></th>
<th>Defines a profile for DSP farm services and enters DSP farm profile configuration mode.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>In the example, the following parameters are set:</td>
</tr>
<tr>
<td><code>Device(config-sccp-ccm)# dspfarm profile 1 transcode universal security</code></td>
<td>• Profile 1 is enabled for transcoding.</td>
</tr>
<tr>
<td></td>
<td>• Profile 1 is enabled for secure DSP farm services.</td>
</tr>
</tbody>
</table>

**Step 9**

<table>
<thead>
<tr>
<th>trustpoint <code>trustpoint-label</code></th>
<th>Associates a trustpoint with a DSP farm profile.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>In the example, the trustpoint to be associated with the DSP farm profile is labeled secdsp.</td>
</tr>
<tr>
<td><code>Device(config-dspfarm-profile)# trustpoint secdsp</code></td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td><strong>Codec or Action</strong></td>
</tr>
<tr>
<td>codec codec-type</td>
<td>Specifies the codecs that are supported by a DSP farm profile.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>Device(config-dspfarm-profile)# codec g711ulaw</code></td>
<td>In the example, the g711ulaw codec is specified.</td>
</tr>
<tr>
<td><strong>Step 11</strong></td>
<td>Repeat Step 10 to configure required codecs.</td>
</tr>
<tr>
<td><strong>Step 12</strong></td>
<td><strong>Maximum Sessions</strong> number</td>
</tr>
<tr>
<td>maximum sessions number</td>
<td>Specifies the maximum number of sessions that are supported by the profile.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>Device(config-dspfarm-profile)# maximum sessions 84</code></td>
<td>In the example, a maximum of 84 sessions are supported by the profile. The maximum number of sessions depends on the number of DSPs available for transcoding.</td>
</tr>
<tr>
<td><strong>Step 13</strong></td>
<td><strong>Associate Application SCCP</strong></td>
</tr>
<tr>
<td>associate application sccp</td>
<td>Associates SCCP to the DSP farm profile.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>Device(config-dspfarm-profile)# associate application sccp</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 14</strong></td>
<td><strong>No Shutdown</strong></td>
</tr>
<tr>
<td>no shutdown</td>
<td>Allocates DSP farm resources and associates them with the application.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>Device(config-dspfarm-profile)# no shutdown</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 15</strong></td>
<td><strong>Exit</strong></td>
</tr>
<tr>
<td>exit</td>
<td>Exits DSP farm profile configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>Device(config-dspfarm-profile)# exit</code></td>
<td></td>
</tr>
</tbody>
</table>

**Registering the Secure Universal Transcoder to the CUBE**

Perform the task in this section to register the secure universal transcoder to the Cisco Unified Border Element. The Cisco Unified Border Element Support for SRTP-RTP Internetworking feature supports both secure transcoders and secure universal transcoders.

**Before You Begin**

Before you register the secure universal transcoder to the Cisco Unified Border Element, you should associate SCCP to the secure DSP farm profile, as described in the **Associating SCCP to the Secure DSP Farm Profile**, on page 220.
SUMMARY STEPS

1. enable
2. configure terminal
3. telephony-service
4. sdspfarm transcode sessions number
5. sdspfarm tag number device-name
6. em logout time1 time2 time3
7. max-ephones max-ephones
8. max-dn max-directory-numbers
9. ip source-address ip-address
10. secure-signaling trustpoint label
11. tftp-server-credentials trustpoint label
12. create cnf-files
13. no sccp
14. sccp
15. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Step 1</th>
<th>enable</th>
<th>Enables privileged EXEC mode.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Example: Device&gt; enable</td>
<td>• Enter your password if prompted.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 2</th>
<th>configure terminal</th>
<th>Enters global configuration mode.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Example: Device&gt; configure terminal</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 3</th>
<th>telephony-service</th>
<th>Enters telephony-service configuration mode.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Example: Device(config)# telephony-service</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 4</th>
<th>sdspfarm transcode sessions number</th>
<th>Specifies the maximum number of transcoding sessions allowed per Cisco CallManager Express router.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Example: Device(config-telephony)# sdspfarm transcode sessions 84</td>
<td>• In the example, a maximum of 84 DSP farm sessions are specified.</td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------</td>
<td>------------------</td>
<td>---------</td>
</tr>
</tbody>
</table>
| **Step 5** | `sdspfarm tag number device-name` | Permits a DSP farm to be registered to Cisco Unified CallManager Express and associates it with an SCCP client interface's MAC address.  
- In the example, DSP farm 1 is associated with the `sxcoder` device. |
| **Step 6** | `em logout time1 time2 time3` | Configures three time-of-day-based timers for automatically logging out all Extension Mobility feature users.  
- In the example, all users are logged out from Extension Mobility after 00:00. |
| **Step 7** | `max-ephones max-ephones` | Sets the maximum number of Cisco IP phones to be supported by a Cisco CallManager Express router.  
- In the example, a maximum of four phones are supported by the Cisco CallManager Express router. |
| **Step 8** | `max-dn max-directory-numbers` | Sets the maximum number of extensions (ephone-dns) to be supported by a Cisco Unified CallManager Express router.  
- In the example, a maximum of four extensions is allowed. |
| **Step 9** | `ip source-address ip-address` | Identifies the IP address and port through which IP phones communicate with a Cisco Unified CallManager Express router.  
- In the example, 10.13.2.52 is configured as the router IP address. |
| **Step 10** | `secure-signaling trustpoint label` | Specifies the name of the Public Key Infrastructure (PKI) trustpoint with the certificate to be used for TLS handshakes with IP phones on TCP port 2443.  
- In the example, PKI trustpoint `secdsp` is configured. |
| **Step 11** | `tftp-server-credentials trustpoint label` | Specifies the PKI trustpoint that signs the phone configuration files.  
- In the example, PKI trustpoint `scme` is configured. |
### Configuring Cisco UBE Support for SRTP-RTP Internetworking

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 12</strong> create cnf-files</td>
<td>Builds the XML configuration files that are required for IP phones in Cisco Unified CallManager Express.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config-telephony)# create cnf-files</td>
<td></td>
</tr>
<tr>
<td><strong>Step 13</strong> no sccp</td>
<td>Disables SCCP and its related applications (transcoding and conferencing) and exits telephony-service configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config-telephony)# no sccp</td>
<td></td>
</tr>
<tr>
<td><strong>Step 14</strong> sccp</td>
<td>Enables SCCP and related applications (transcoding and conferencing).</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config)# sccp</td>
<td></td>
</tr>
<tr>
<td><strong>Step 15</strong> end</td>
<td>Exits global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config)# end</td>
<td></td>
</tr>
</tbody>
</table>

**Configuring SRTP-RTP Internetworking Support**

Perform the task in this section to enable SRTP-RTP internetworking support between one or multiple Cisco Unified Border Elements for SIP-SIP audio calls. In this task, RTP is configured on the incoming call leg and SRTP is configured on the outgoing call leg.

**Before You Begin**

Before you configure the Cisco Unified Border Element Support for SRTP-RTP Internetworking feature, you should register the secure universal transcoder to the Cisco Unified Border Element, as described in the **Registering the Secure Universal Transcoder to the CUBE, on page 223.**

**Note**

The Cisco Unified Border Element Support for SRTP-RTP Internetworking feature is available only on platforms that support transcoding on the Cisco Unified Border Element. The feature is also available only on secure Cisco IOS images on the Cisco Unified Border Element.

>
### SUMMARY STEPS

1. enable  
2. configure terminal  
3. dial-peer voice tag voip  
4. destination-pattern string  
5. session protocol sipv2  
6. session target ipv4: destination-address  
7. incoming called-number string  
8. codec codec  
9. end  
10. dial-peer voice tag voip  
11. Repeat Steps 4, 5, 6, and 7 to configure a second dial peer.  
12. srtp  
13. codec codec  
14. exit

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>enable</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device&gt; enable</td>
</tr>
<tr>
<td></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>configure terminal</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device# configure terminal</td>
</tr>
<tr>
<td></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>dial-peer voice tag voip</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config)# dial-peer voice 201 voip</td>
</tr>
<tr>
<td></td>
<td>Defines a particular dial peer, to specify the method of voice encapsulation, and enters dial peer voice configuration mode.</td>
</tr>
<tr>
<td></td>
<td>• In the example, the following parameters are set:</td>
</tr>
<tr>
<td></td>
<td>• Dial peer 201 is defined.</td>
</tr>
<tr>
<td></td>
<td>• VoIP is shown as the method of encapsulation.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>destination-pattern string</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device(config-dial-peer)# destination-pattern 5550111</td>
</tr>
<tr>
<td></td>
<td>Specifies either the prefix or the full E.164 telephone number to be used for a dial peer string.</td>
</tr>
<tr>
<td></td>
<td>• In the example, 5550111 is specified as the pattern for the telephone number.</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Specifies a session protocol for calls between local and remote routers using the packet network.</td>
</tr>
<tr>
<td><code>session protocol sipv2</code></td>
<td>- In the example, the <code>sipv2</code> keyword is configured so that the dial peer uses the IETF SIP.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Device(config-dial-peer)# session protocol sipv2</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>Designates a network-specific address to receive calls from a VoIP or VoIP v6 dial peer.</td>
</tr>
<tr>
<td><code>session target ipv4: destination-address</code></td>
<td>- In the example, the IP address of the dial peer to receive calls is configured as 10.13.25.102.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Device(config-dial-peer)# session target ipv4:10.13.25.102</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>Specifies a digit string that can be matched by an incoming call to associate the call with a dial peer.</td>
</tr>
<tr>
<td><code>incoming called-number string</code></td>
<td>- In the example, 5550111 is specified as the pattern for the E.164 or private dialing plan telephone number.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Device(config-dial-peer)# incoming called-number 5550111</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>Specifies the voice coder rate of speech for the dial peer.</td>
</tr>
<tr>
<td><code>codec codec</code></td>
<td>- In the example, G.711 mu-law at 64,000 bps, is specified as the voice coder rate for speech.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Device(config-dial-peer)# codec g711ulaw</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td>Exits dial peer voice configuration mode.</td>
</tr>
<tr>
<td><code>end</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Device(config-dial-peer)#end</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td>Defines a particular dial peer, to specify the method of voice encapsulation, and enters dial peer voice configuration mode.</td>
</tr>
<tr>
<td><code>dial-peer voice tag voip</code></td>
<td>- In the example, the following parameters are set:</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Device(config)# dial-peer voice 200 voip</code></td>
<td>- Dial peer 200 is defined.</td>
</tr>
<tr>
<td></td>
<td>- VoIP is shown as the method of encapsulation.</td>
</tr>
<tr>
<td><strong>Step 11</strong></td>
<td>Repeat Steps 4, 5, 6, and 7 to configure a second dial peer.</td>
</tr>
<tr>
<td>--</td>
<td></td>
</tr>
<tr>
<td><strong>Step 12</strong></td>
<td>Specifies that SRTP is used to enable secure calls for the dial peer.</td>
</tr>
<tr>
<td><code>srtp</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Device(config-dial-peer)# srtp</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 13</strong></td>
<td>Specifies the voice coder rate of speech for the dial peer.</td>
</tr>
<tr>
<td><code>codec codec</code></td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device(config-dial-peer)# codec g711ulaw</td>
<td>• In the example, G.711 mu-law at 64,000 bps, is specified as the voice coder rate for speech.</td>
</tr>
<tr>
<td><strong>Step 14</strong></td>
<td>exit</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device(config-dial-peer)# exit</td>
<td></td>
</tr>
</tbody>
</table>

**Troubleshooting Tips**

The following commands can help troubleshoot Cisco Unified Border Element support for SRTP-RTP internetworking:

- show crypto pki certificates
- show sccp
- show sdspfarm

**Enabling SRTP on the Cisco UBE**

You can configure SRTP with the fallback option so that a call can fall back to RTP if SRTP is not supported by the other call end. Enabling SRTP is required for supporting nonsecure supplementary services such as MoH, call forward, and call transfer.

**Enabling SRTP Globally**

Perform this task to enable SRTP globally.

**SUMMARY STEPS**

1. enable
2. configure terminal
3. voice service voip
4. srtp fallback
5. exit
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td>• Enter your password if prompted.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> voice service voip</td>
<td>Enters voice-service configuration mode and specifies VoIP encapsulation as the voice-encapsulation type.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device(config)# voice service voip</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> srtp fallback</td>
<td>Enables call fallback to nonsecure mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Note If the secure SIP trunk is towards the Cisco UCM, you must configure the srtp negotiate cisco command in voice-service configuration mode for a non-Cisco fallback to work.</td>
</tr>
<tr>
<td>RoDeviceuter(conf-voi-serv)# srtp fallback</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> exit</td>
<td>Exits voice service configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device(conf-voi-serv)# exit</td>
<td></td>
</tr>
</tbody>
</table>

### Example: Enabling SRTP Globally

```
Device(config)# voice service voip
Device(config)# srtp fallback
Device(config)# exit
```

### Enabling SRTP on a Dial Peer

Perform this task to enable SRTP on a dial peer.
SUMMARY STEPS

1. enable
2. configure terminal
3. dial-peer voice tag voip
4. srtp fallback
5. exit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>enable</td>
</tr>
<tr>
<td>Example:</td>
<td>Device&gt; enable</td>
</tr>
<tr>
<td></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>configure terminal</td>
</tr>
<tr>
<td>Example:</td>
<td>Device# configure terminal</td>
</tr>
<tr>
<td></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>dial-peer voice tag voip</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config)# dial-peer voice 10 voip</td>
</tr>
<tr>
<td></td>
<td>Defines a particular dial peer to specify VoIP as the method of voice encapsulation and enters dial peer voice configuration mode.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>srtp fallback</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config-dial-peer)# srtp fallback</td>
</tr>
<tr>
<td></td>
<td>Enables specific dial-peer calls to fall back to nonsecure mode.</td>
</tr>
<tr>
<td>Note</td>
<td>If the secure SIP trunk is towards the Cisco UCM, you must configure the srtp negotiate cisco command in dial peer voice configuration mode for a non-Cisco fallback to work.</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>exit</td>
</tr>
<tr>
<td>Example:</td>
<td>Device(config-dial-peer)# exit</td>
</tr>
<tr>
<td></td>
<td>Exits dial peer voice configuration mode.</td>
</tr>
</tbody>
</table>

Example: Enabling SRTP on a Dial Peer

```
Device(config)# dial-peer voice 10 voip
Device(config-dial-peer)# srtp fallback
Device(config-dial-peer)# exit
```
Troubleshooting Tips

The following commands can help troubleshoot SRTP-RTP supplementary services support on Cisco UBE:

- debug ccsip all
- debug sccp all
- debug voip ccapi inout

Verifying SRTP-RTP Supplementary Services Support on the Cisco UBE

Perform this task to verify the configuration for SRTP-RTP supplementary services support on the Cisco UBE. The show commands need not be entered in any specific order.

SUMMARY STEPS

1. enable
2. show call active voice brief
3. show sccp connection
4. show dspfarm dsp active

DETAILED STEPS

Step 1 enable
Enables privileged EXEC mode.

Example:

Device> enable

Step 2 show call active voice brief
Displays call information for voice calls in progress.

Example:

Device# show call active voice brief
Telephony call-legs: 0
SIP call-legs: 2
H323 call-legs: 0
Call agent controlled call-legs: 0
SCCP call-legs: 2
ulticast call-legs: 0
Total call-legs: 4
0 : 1 12:49:45.256 IST Fri Jun 3 2011.1 +29060 pid:1 Answer 10008001 connected
dur 00:01:19 tx:1653/271092 rx:2831/464284 dscp:0 media:0
IP 10.45.40.40:7892 SRTP: on rtt:0ms pl:0/0ms lost:0/0/0 delay:0/0/0ms g711ulaw TextRelay: off
media inactive detected:n media contrl rcvd:n/a timestamp:n/a long duration call detected:n long duration call duration:n/a timestamp:n/a

0 : 2 12:49:45.256 IST Fri Jun 3 2011.2 +29060 pid:22 Originante 20009001 connected
dur 00:01:19 tx:2831/452960 rx:1653/264480 dscp:0 media:0
IP 10.45.40.40:7893 SRTP: off rtt:0ms pl:0/0ms lost:0/0/0 delay:0/0/0ms g711ulaw TextRelay: off
media inactive detected:n media contrl rcvd:n/a timestamp:n/a
Step 3  show sccp connection
Displays SCCP connection details.

Example:

```
Device# show sccp connection
sess_id conn_id stype mode codec sport rport ripaddr conn_id_tx
65537 4 s-xcode sendrecv g711u 17124 2000 10.45.34.252
65537 8 xcode sendrecv g711u 30052 2000 10.45.34.252
```

Total number of active session(s) 1, and connection(s) 2

Step 4  show dspfarm dsp active
Displays active DSP information about the DSP farm service.

Example:

```
Device# show dspfarm dsp active
SLOT DSP VERSION STATUS CHNL USE TYPE RSC_ID BRIDGR_ID PKTS_TXED PKTS_RXED
0 1 30.0.209 UP 1 USED xcode 1 4 2876 1706
0 1 30.0.209 UP 1 USED xcode 1 5 1698 2876
```

Total number of DSPFARM DSP channel(s) 1

---

**Configuration Examples for CUBE Support for SRTP-RTP Internetworking**

**SRTP-RTP Internetworking Example**

The following example shows how to configure Cisco Unified Border Element support for SRTP-RTP internetworking. In this example, the incoming call leg is RTP and the outgoing call leg is SRTP.

```
enable
configure terminal
ip http server
crypto pki server 3845-cube
database level complete
```
grant auto
no shutdown
%PKI-6-CS_GRANT_AUTO: All enrollment requests will be automatically granted.
% Some server settings cannot be changed after CA certificate generation.
% Please enter a passphrase to protect the private key or type Return to exit
Password:
Re-enter password:
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]
% SSH-5-ENABLED: SSH 1.99 has been enabled
% Exporting Certificate Server signing certificate and keys...
% Certificate Server enabled.
%PKI-6-CS_ENABLED: Certificate server now enabled.

crypto pki trustpoint secdsp
enrollment url http://10.13.2.52:80
serial-number
revocation-check crl
rsakeypair 3845-cube
exit

crypto pki authenticate secdsp
Certificate has the following attributes:
  Fingerprint MD5: CCC82E9E 4382CCFE ADA0EB8C 524E2FC1
  Fingerprint SHA1: 34B9C4BF 4841AB31 7B0810AD 80084475 3965F140
% Do you accept this certificate? [yes/no]: yes
Trustpoint CA certificate accepted.
crypto pki enroll secdsp
% Start certificate enrollment ..
% Create a challenge password. You will need to verbally provide this password to the CA Administrator in order to revoke your certificate. For security reasons your password will not be saved in the configuration. Please make a note of it.
Password:
Re-enter password:
% The subject name in the certificate will include: 3845-CUBE
% The serial number in the certificate will be: FHK1212F4MU
% Include an IP address in the subject name? [no]:
% Request certificate from CA? [yes/no]: yes
% Certificate request sent to Certificate Authority
% The 'show crypto pki certificate secdsp verbose' command will show the fingerprint.
CRYPTO_PKI: Certificate Request Fingerprint MD5: 56CE5FC3 B8411CF3 93A343DA 785C2360
CRYPTO_PKI: Certificate Request Fingerprint SHA1: EE029629 55F5CA10 21E50F08 F56440A2
%PKI-6-CERTRET: Certificate received from Certificate Authority

voice-card 0
dspfarm
dsp services dspfarm
voice-card 1
dspfarm
dsp services dspfarm
exit

sccp local GigabitEthernet 0/0
sccp ccm 10.13.2.52 identifier 1 version 5.0.1
sc cops
SCCP operational state bring up is successful.sccp ccm group 1
associate ccm 1 priority 1
associate profile 1 register sxcoder
dspfarm profile 1 transcode universal security
trustpoint secdsp
codec g711ulaw
codec g711alaw
codec g729ar8
codec g729abr8
codec g729r8
codec ilbc
codec g729br8
maximum sessions 84
associate application sccp
no shutdown
exit

telephony-service
Updating CNF files

CNF-FILES: Clock is not set or synchronized, retaining old versionStamps

CNF files updating complete

secure-signaling trustpoint secdsp
tftp-server-credentials trustpoint scme

CNF-FILES: Clock is not set or synchronized, retaining old versionStamps

create cnf-files

CNF-FILES: Clock is not set or synchronized, retaining old versionStamps

no sccp

! sccp

SCCP operational state bring up is successful.

end

%SDSPFARM-6-REGISTER: mtp-1:sxcoder IP:10.13.2.52 Socket:1 DeviceType:MTP has registered.

%SYS-5-CONFIG_I: Configured from console by console
dial-peer voice 201 voip
destination-pattern 5550111
session protocol sipv2
session target ipv4:10.13.25.102
incoming called-number 5550112
codec g711ulaw

! dial-peer voice 200 voip
destination-pattern 5550112
session protocol sipv2
session target ipv4:10.13.2.51
incoming called-number 5550111
srtp
codec g711ulaw

Feature Information for CUBE Support for SRTP-RTP Internetworking

Table 22: Feature Information for Cisco Unified Border Element Support for SRTP-RTP Internetworking

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco Unified Border Element Support for SRTP-RTP Internetworking</td>
<td>12.4(22)YB, 15.0(1)M</td>
<td>This feature allows secure enterprise-to-enterprise calls. Support for SRTP-RTP internetworking between one or multiple Cisco Unified Border Elements is enabled for SIP-SIP audio calls. The following sections provide information about this feature: The following command was introduced: <code>tls</code>.</td>
</tr>
</tbody>
</table>

Cisco Unified Border Element Protocol-Independent Features and Setup Configuration Guide, Cisco IOS Release 15M&T
### Feature Information

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supplementary Services Support on Cisco UBE for RTP-SRTP Calls</td>
<td>15.2(1)T</td>
<td>The SRTP-RTP Internetworking feature was enhanced to support supplementary services for SRTP-RTP calls on Cisco UBE.</td>
</tr>
<tr>
<td>Supplementary Services Support on Cisco UBE for RTP-SRTP Calls</td>
<td>Cisco IOS XE Release 3.7S</td>
<td>The SRTP-RTP Internetworking feature was enhanced to support supplementary services for SRTP-RTP calls on Cisco UBE.</td>
</tr>
</tbody>
</table>
Support for SRTP Termination

This Support for SRTP Termination feature enables Cisco Unified Border Element (Cisco UBE) support for Secure Real-time Transport Protocol (SRTP) on the Session Initiation Protocol (SIP) Trunk interface.

- Finding Feature Information, page 237
- Information About Support for SRTP Termination, page 237
- How to Configure Support for SRTP Termination, page 240
- Verifying Support for SRTP Termination, page 242
- Configuration Examples for Support for SRTP Termination, page 243
- Additional References for Support for SRTP Termination, page 244
- Feature Information for Support for SRTP Termination, page 244

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About Support for SRTP Termination

The Support for SRTP Termination feature configures Cisco Unified Border Element (Cisco UBE) support for an Secure Real-time Transport Protocol (SRTP) connection using the AES_CM_128_HMAC_SHA1_80 crypto suite. This feature implements crypto-suite negotiation and appropriately sets up the call on the following two sides:

- The Cisco Unified Call Manager (CUCM) or IP phones side—Connection between the end devices and CUBE
• SIP Trunk side—Connection between CUBE and Service Provider

Prior to the Support for SRTP Termination feature, Cisco UBE could support an SRTP connection using the AES_CM_128_HMAC_SHA1_32 crypto suite. This crypto suite is still used by default, unless Cisco UBE is configured to use AES_CM_128_HMAC_SHA1_80 crypto suite.

Cisco UBE SRTP termination can be implemented in the following ways:

• SRTP-RTP interworking—This method is used with devices (CUCM or IP Phone devices) that still support AES_CM_128_HMAC_SHA1_32 crypto suite only.

• SRTP-SRTP pass-through—This method is used with devices that support AES_CM_128_HMAC_SHA1_80 crypto suite.

Note: This method of implementation is currently supported by non-CUCM end devices like Microsoft Link. This method can also be used when CUCM or IP phone devices support AES_CM_128_HMAC_SHA1_80 crypto suite.

For End Devices Supporting AES_CM_128_HMAC_SHA1_80 Crypto Suite

This method is used between Cisco Unified Border Element (Cisco UBE), IP Phones, and other Cisco Unified Call Manager (CUCM) devices that support AES_CM_128_HMAC_SHA1_80 crypto suite.

• CUCM or IP Phones side—A Secure Real-time Transport Protocol (SRTP) connection using the AES_CM_128_HMAC_SHA1_80 crypto suite exists here. In the figure below, IP Phone and CUBE within the customer network connect with an SRTP connection using AES_CM_128_HMAC_SHA1_80 crypto suite.

• Session Initiation Protocol (SIP) Trunk side—An SRTP connection using the AES_CM_128_HMAC_SHA1_80 crypto suite. In the figure below, CUBE on the Customer Network
and SBC on the Service Provider Network connect with an SRTP connection using the AES_CM_128_HMAC_SHA1_80 crypto suite.

**Figure 15: SRTP Connection Supporting AES_CM_128_HMAC_SHA1_80 crypto suite**

For End Devices Supporting AES_CM_128_HMAC_SHA1_32 Crypto Suite

A single Cisco Unified Call Manager (Cisco UBE) device cannot terminate a Secure Real-time Transport Protocol (SRTP) connection with an IP Phone using the AES_CM_128_HMAC_SHA1_32 crypto suite and initiate an SRTP connection with an external Cisco UBE device with the AES_CM_128_HMAC_SHA1_80 crypto suite at the same time.

For Cisco Unified Call Manager (CUCM) and IP Phone devices that support only AES_CM_128_HMAC_SHA1_32 crypto suite, the interim SRTP-RTP interworking solution that is described below can be implemented.

- CUCM or IP Phone side:
  - An SRTP connection using the AES_CM_128_HMAC_SHA1_32 crypto suite exists between the IP Phone and CUBE1.
  - An RTP connection exists between CUBE1 and CUBE2.
• SIP trunk side—An SRTP connection using the AES_CM_128_HMAC_SHA1_80 crypto suite is initiated by CUBE2 here. In the image below, CUBE2 is the border element on the Customer Network and SBC is the border element on the Service Provider Network.

Figure 16: SRTP-RTP Interworking Supporting AES_CM_128_HMAC_SHA1_32 crypto suite

How to Configure Support for SRTP Termination

Configuring Crypto Authentication

Configuring Crypto Authentication (Global Level)

SUMMARY STEPS

1. enable
2. configure terminal
3. voice service voip
4. sip
5. srtp-auth {sha1-32 | sha1-80}
6. end
DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example: Device&gt; enable</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Step 2 configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 3 voice service voip</td>
<td>Specifies VoIP encapsulation and enters voice-service configuration mode.</td>
</tr>
<tr>
<td>Example: Device(config)# voice service voip</td>
<td></td>
</tr>
<tr>
<td>Step 4 sip</td>
<td>Enters the Session Initiation Protocol (SIP) configuration mode.</td>
</tr>
<tr>
<td>Example: Device(conf-voi-serv)# sip</td>
<td></td>
</tr>
<tr>
<td>Step 5 srtp-auth {sha1-32</td>
<td>Configures an SRTP connection on CUBE using the preferred crypto suite.</td>
</tr>
<tr>
<td></td>
<td>sha1-80}</td>
</tr>
<tr>
<td>Example: Device(conf-serv-sip)# srtp-auth sha1-80</td>
<td>• The default value is sha1-32.</td>
</tr>
<tr>
<td>Step 6 end</td>
<td>Ends the current configuration session and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td>Example: Router(conf-serv-sip)# end</td>
<td></td>
</tr>
</tbody>
</table>

Configuring Crypto Authentication (Dial Peer Level)

SUMMARY STEPS

1. enable
2. configure terminal
3. dial-peer voice tag voip
4. voice-class sip srtp-auth {sha1-32 | sha1-80 | system}
5. end
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>dial-peer voice <em>tag</em> voip</td>
<td>Defines a VoIP dial peer and enters dial peer voice configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> Device(config)# dial-peer voice 15 voip</td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>voice-class sip srtp-auth {sha1-32</td>
<td>sha1-80</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> Device(config-dial-peer)# voice-class sip srtp-auth sha1-80</td>
<td>- The default value is <strong>sha1-32</strong>.</td>
</tr>
<tr>
<td>Step 5</td>
<td>end</td>
<td>Ends the current configuration session and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> Router(conf-serv-sip)# end</td>
<td></td>
</tr>
</tbody>
</table>

### Verifying Support for SRTP Termination

Perform this task to verify the configuration of an SRTP connection on Cisco Unified Border Element using the AES_CM_128_HMAC_SHA1_80 crypto suite. The show commands can be entered in any order.

### SUMMARY STEPS

1. show sip-ua calls
2. show sip-ua srtp

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>show sip-ua calls</td>
<td></td>
</tr>
</tbody>
</table>
The following example displays sample output for active user agent client (UAC) and user agent server (UAS) information on Session Initiation Protocol (SIP) calls:

Device# show sip-ua calls
Call 1
 SIP Call ID : 20894
  Media Stream 1
    Local Crypto Suite : AES_CM_128_HMAC_SHA1_80
    Remote Crypto Suite: AES_CM_128_HMAC_SHA1_80 (AES_CM_128_HMAC_SHA1_80 AES_CM_128_HMAC_SHA1_32)

Step 2 show sip-ua srtp

Example:
The following example displays sample output for Session Initiation Protocol (SIP) user-agent (UA) SRTP information:

Device# show sip-ua srtp
SIP UA SRTP
 Crypto-suite Negotiation
 AES_CM_128_HMAC_SHA1_80: 3
 AES_CM_128_HMAC_SHA1_32: 2

Configuration Examples for Support for SRTP Termination

Example: Configuring Crypto Authentication

Example: Configuring Crypto Authentication (Global Level)
The following example shows how to configure Cisco UBE to support an SRTP connection using the AES_CM_128_HMAC_SHA1_80 crypto suite at the global level:

Device> enable
Device# configure terminal
Device(config)# voice service voip
Device(config-voip)# sip
Device(config-serv-sip)# srtp-auth sha1-80
Device(config-serv-sip)# end

Example: Configuring Crypto Authentication (Dial Peer Level)
The following example shows how to configure Cisco UBE to support an SRTP connection using the AES_CM_128_HMAC_SHA1_80 crypto suite at the dial peer level:

Device> enable
Device# configure terminal
Device(config)# dial-peer voice 15 voip
Device(config-dial-peer)# voice-class sip srtp-auth sha1-80
Device(config-dial-peer)# end
Additional References for Support for SRTP Termination

Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voice commands</td>
<td>Cisco IOS Voice Command Reference</td>
</tr>
<tr>
<td>Cisco IOS commands</td>
<td>Cisco IOS Master Command List, All Releases</td>
</tr>
<tr>
<td>SIP configuration tasks</td>
<td>SIP Configuration Guide, Cisco IOS Release 15M&amp;T</td>
</tr>
</tbody>
</table>

Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
<td><a href="http://www.cisco.com/support">http://www.cisco.com/support</a></td>
</tr>
</tbody>
</table>

Feature Information for Support for SRTP Termination

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 . An account on Cisco.com is not required.
Table 23: Feature Information for Support for SRTP Termination

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support for SRTP Termination</td>
<td>15.4(1)T</td>
<td>The Support for SRTP Termination feature describes how to configure Cisco Unified Border Element to support AES_CM_128_HMAC_SHA1_80 crypto suite on the Session Initiation Protocol (SIP) Trunk interface. The following commands were introduced or modified: <em>show sip-ua srt.</em>, <em>srt-auth</em> and <em>voice-class sip srt-auth</em>.</td>
</tr>
</tbody>
</table>
CHAPTER 25

Configuring RTCP Report Generation

The assisted Real-time Transport Control Protocol (RTCP) feature adds the ability for Cisco Unified Border Element (Cisco UBE) to generate standard RTCP keepalive reports on behalf of endpoints. RTCP reports determine the liveliness of a media session during prolonged periods of silence, such as call hold or mute. Therefore, it is important for the Cisco UBE to generate RTCP reports irrespective of whether the endpoints send or receive media.

Cisco UBE generates RTCP report only when inbound and outbound call legs are SIP, or SIP to H.323, or H.323 to SIP.

- Finding Feature Information, page 247
- Prerequisites, page 248
- Restrictions, page 248
- Configuring RTCP Report Generation on Cisco UBE, page 248
- Troubleshooting Tips, page 250
- Feature Information for Configuring RTCP Report Generation, page 251

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.
Prerequisites

Cisco Unified Border Element

- Cisco IOS Release 15.1(2)T or a later release must be installed and running on your Cisco Unified Border Element.

Cisco Unified Border Element (Enterprise)

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

Restrictions

- RTCP report generation over IPv6 is not supported.
- RTCP report generation is not supported for Secure Real-time Transport Protocol (SRTP) or SRT Control Protocol (SRTCP) pass-through as Cisco UBE is not aware of the media encryption or decryption keys.
- RTCP report generation is not supported for loopback calls, T.38 fax, and modem relay calls.
- RTCP or SRTCP report generation is not supported when Cisco UBE inserts a Digital Signal Processor (DSP) for RTP-SRTP interworking on RTP and SRTP call legs.
- RTCP report generation is not supported when there is a call hold with an invalid media address such as 0.0.0.0 in Session Description Protocol (SDP) or Open Logical Channel (OLC).
- RTCP report generation is not supported for RTCP multiplexed with RTP on the same address and port.
- RTCP report generation is not supported on enterprise aggregation services routers (ASR) Cisco UBE.
- RTCP packet generation is not supported on the SIP leg when the H.323 leg puts the SIP leg on hold in a Slow Start to Delayed-Offer call.

Configuring RTCP Report Generation on Cisco UBE

RTCP keepalive packets indicate session liveliness. When configured on Cisco UBE, RTCP keepalive packets are sent on both inbound and outbound SIP or H.323 call legs.

Perform this task to configure RTCP report generation on Cisco UBE.
### SUMMARY STEPS

1. enable
2. configure terminal
3. voice service voip
4. allow-connections from-type to to-type
5. rtp keepalive
6. end

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router&gt; enable</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Step 2</td>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>voice service voip</td>
<td>Enters voice service configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config)# voice service voip</td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>allow-connections from-type to to-type</td>
<td>Allows connections between SIP endpoints in a VoIP network.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(conf-voi-serv)# allow-connections sip to sip</td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td>rtp keepalive</td>
<td>Configures RTCP keepalive report generation.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(conf-voi-serv)# rtp keepalive</td>
<td></td>
</tr>
<tr>
<td>Step 6</td>
<td>end</td>
<td>Exits voice service configuration mode and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(conf-voi-serv)# end</td>
<td></td>
</tr>
</tbody>
</table>
Troubleshooting Tips

Use the following debug commands for debugging related to RTCP keepalive packets:

• **debug voip rtcp packet** -- Shows details related to RTCP keepalive packets such as RTCP sending and receiving paths, Call ID, Globally Unique Identifier (GUID), packet header, and so on.

Router# `debug voip rtcp packet`
01:06:27.450: //6/xxxxxxxxxxxx/RTP//Event/voip_rtp_send_rtcp_keepalive: Generate RTCP Keepalive
*Mar 17 01:06:27.450: rtcp_send_report: Attributes
  (src ip=192.168.30.3, src port=17101, dst ip=192.168.30.4, dst port=18619
  bye=0, initial=1, ssrc=0x07111E02, keepalive=1)
  (rtcp=0x2E5AF214, ssrc=0x07111E02, source->ssrc=0x00001E03, total_len=36)
  2E5AF210: 80C90001 07111E02 81CA0006 .I.......J..
  2E5AF220: 07111E02 010F302E 302E3040 392E3435 ......0.0.0@9.45
  2E5AF230: 2E33302E 33000000 00 .30.3....

Caution

Under moderate traffic loads, the **debug voip rtp packet** command produces a high volume of output and the command should be enabled only when the call volume is very low.

• **debug voip rtp packet** -- Shows details about VoIP RTP packet debugging trace.

Router# `debug voip rtp packet`
VOIP RTP All Packets debugging is on

• **debug voip rtp session** -- Shows all RTP session debug information.

Router# `debug voip rtp session`
VOIP RTP All Events debugging is on

• **debug voip rtp error** -- Shows details about debugging trace for RTP packet error cases.

Router# `debug voip rtp error`
VOIP RTP Errors debugging is on

• **debug ip rtp protocol** -- Shows details about RTP protocol debugging trace.

Router# `debug ip rtp protocol`
RTP protocol debugging is on

• **debug voip rtcp session** -- Shows all RTCP session debug information.

Router# `debug voip rtcp session`
VOIP RTCP Events debugging is on

• **debug voip rtcp error** -- Shows details about debugging trace for RTCP packet error cases.

Router# `debug voip rtcp error`
VOIP RTCP Errors debugging is on
Feature Information for Configuring RTCP Report Generation

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 . An account on Cisco.com is not required.

Feature History Table entry for the Cisco Unified Border Element.

Table 24: Feature Information for Configuring RTCP Report Generation

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assisted RTCP</td>
<td>15.1(2)T</td>
<td>Assisted RTCP Feature Information for Configuring RTCP Report Generation This feature adds the ability for Cisco UBE to generate standard RTCP keepalive reports on behalf of endpoints and ensures the liveliness of a media session during prolonged periods of silence, such as call hold. The following commands were introduced or modified in this release: <code>rtp keepalive</code>, <code>debug voip rtp</code>, <code>debug voip rtp</code>, <code>debug ip rtp protocol</code>, and <code>ip rtcp report interval</code>.</td>
</tr>
</tbody>
</table>

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

Table 25: Feature Information for Configuring RTCP Report Generation

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assisted RTCP</td>
<td>TBD</td>
<td>Assisted RTCP Feature Information for Configuring RTCP Report Generation This feature adds the ability for Cisco UBE to generate standard RTCP keepalive reports on behalf of endpoints and ensures the liveliness of a media session during prolonged periods of silence, such as call hold. The following commands were introduced or modified in this release: <code>rtp keepalive</code>, <code>debug voip rtp</code>, <code>debug voip rtp</code>, <code>debug ip rtp protocol</code>, and <code>ip rtcp report interval</code>.</td>
</tr>
</tbody>
</table>
CHAPTER 26

SIP SRTP Fallback to Nonsecure RTP

The SIP SRTP Fallback to Nonsecure RTP feature enables a Cisco IOS Session Initiation Protocol (SIP) gateway to fall back from Secure Real-time Transport Protocol (SRTP) to Real-time Transport Protocol (RTP) by accepting or sending an RTP/Audio-Video Profile (AVP) (RTP) profile in response to an RTP/SAVP (SRTP) profile. This feature also allows inbound and outbound SRTP calls with nonsecure SIP signaling schemes (such as SIP URL) and provides the administrator the flexibility to configure Transport Layer Security (TLS), IPsec, or any other security mechanism used in the lower layers for secure signaling of crypto attributes.

- Finding Feature Information, page 253
- Prerequisites for SIP SRTP Fallback to Nonsecure RTP, page 253
- Configuring SIP SRTP Fallback to Nonsecure RTP, page 254
- Feature Information for SIP SRTP Fallback to Nonsecure RTP, page 254

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for SIP SRTP Fallback to Nonsecure RTP

Cisco Unified Border Element

- Cisco IOS Release 12.4(22)T 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 SRTP Fallback to Nonsecure RTP

To enable this feature, see the "Configuring SIP Support for SRTP" section of the Cisco IOS SIP Configuration Guide, Release 15.1 at the following URL:

Detailed command information for the `srtp`, `srtp negotiate`, and `voice-class sip srtp negotiate` commands is located in the Cisco IOS Voice Command Reference

### Feature Information for SIP SRTP Fallback to Nonsecure RTP

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 . An account on Cisco.com is not required.

Table 26: Feature Information for SIP SRTP Fallback to Nonsecure RTP

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIP SRTP Fallback to Nonsecure RTP</td>
<td>12.4(22)T</td>
<td>The SIP SRTP Fallback to Nonsecure RTP feature enables a Cisco IOS Session Initiation Protocol (SIP) gateway to fall back from SRTP to RTP by accepting or sending an RTP/AVP(RTP) profile in response to an RTP/SAVP(SRTP) profile. This feature also allows inbound and outbound SRTP calls with nonsecure SIP signaling schemes (such as SIP URL) and provides the administrator the flexibility to configure TLS, IPsec, or any other security mechanism used in the lower layers for secure signaling of crypto attributes. The following commands were introduced or modified: <code>srtp</code> (voice), <code>srtp negotiate</code>, and <code>voice-class sip srtp negotiate</code></td>
</tr>
</tbody>
</table>

---

Cisco Unified Border Element Protocol-Independent Features and Setup Configuration Guide, Cisco IOS Release 15M&T
<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIP SRTP Fallback to Nonsecure RTP</td>
<td>Cisco IOS XE Release 3.1S</td>
<td>The SIP SRTP Fallback to Nonsecure RTP feature enables a Cisco IOS Session Initiation Protocol (SIP) gateway to fall back from SRTP to RTP by accepting or sending an RTP/AVP(RTP) profile in response to an RTP/SAVP(SRTP) profile. This feature also allows inbound and outbound SRTP calls with nonsecure SIP signaling schemes (such as SIP URL) and provides the administrator the flexibility to configure TLS, IPsec, or any other security mechanism used in the lower layers for secure signaling of crypto attributes. The following commands were introduced or modified: <code>srtp (voice)</code>, <code>srtp negotiate</code>, and <code>voice-class sip srtp negotiate</code></td>
</tr>
</tbody>
</table>
Feature Information for SIP SRTP Fallback to Nonsecure RTP
Configuring Support for Interworking Between RSVP Capable and RSVP Incapable Networks

The Support for Interworking Between RSVP Capable and RSVP Incapable Networks feature provides precondition-based Resource Reservation Protocol (RSVP) support for basic audio call and supplementary services on Cisco UBE. This feature improves the interoperability between RSVP and non-RSVP networks. RSVP functionality added to Cisco UBE helps you to reserve the required bandwidth before making a call. This feature extends RSVP support to delayed-offer to delayed-offer and delayed-offer to early-offer calls, along with the early-offer to early-offer calls.

- Finding Feature Information, page 258
- Prerequisites, page 258
- Restrictions, page 258
- Configuring RSVP on an Interface, page 258
- Configuring Optional RSVP on the Dial Peer, page 259
- Configuring EO to EO DO to DO and DO to EO at the Dial Peer, page 261
- Configuring Mandatory RSVP on the Dial Peer, page 263
- Configuring Midcall RSVP Failure Policies, page 264
- Configuring DSCP Values, page 266
- Configuring an Application ID, page 267
- Configuring Priority, page 268
- Troubleshooting the Support for Interworking Between RSVP Capable and RSVP Incapable Networks Feature, page 269
- Verifying Support for Interworking Between RSVP Capable and RSVP Incapable Networks, page 270
- Feature Information for Configuring Support for Interworking Between RSVP Capable and RSVP Incapable Networks, page 271
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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites

- RSVP policies allow you to configure separate bandwidth pools with varying limits so that any one application, such as video, can consume all the RSVP bandwidth on a specified interface at the expense of other applications, such as voice, which would be dropped.

- To limit bandwidth per application, you must configure a bandwidth limit before configuring Support for the Interworking Between RSVP Capable and RSVP Incapable Networks feature. See the Configuring RSVP on an Interface task.

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 <TBD> or a later release must be installed and running on your Cisco ASR 1000 Series Router.

Restrictions

The Support for Interworking Between RSVP Capable and RSVP Incapable Networks feature has the following restrictions:

- Segmented RSVP is not supported.

- Interoperability between Cisco UBE and Cisco Unified Communications Manager is not available.

- RSVP-enabled video calls are not supported.

Configuring RSVP on an Interface

You must allocate some bandwidth for the interface before enabling RSVP. Perform this task to configure RSVP on an interface.
SUMMARY STEPS

1. enable
2. configure terminal
3. interface type slot / port
4. ip rsvp bandwidth [reservable-bw [max-reservable-bw] [sub-pool reservable-bw]]
5. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>enable</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>configure terminal</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>interface type slot / port</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config)# interface FastEthernet 0/1</td>
<td>Configures an interface type and enters interface configuration mode.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>ip rsvp bandwidth [reservable-bw [max-reservable-bw] [sub-pool reservable-bw]]</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-if)# ip rsvp bandwidth 10000 100000</td>
<td>Enables RSVP for IP on an interface.</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>end</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-if)# end</td>
<td>(Optional) Exits interface configuration mode and returns to privileged EXEC mode.</td>
</tr>
</tbody>
</table>

Configuring Optional RSVP on the Dial Peer

Perform this task to configure optional RSVP at the dial peer level. This configuration allows you to have uninterrupted call even if there is a failure in bandwidth reservation.
SUMMARY STEPS

1. enable
2. configure terminal
3. dial-peer voice tag voip
4. no acc-qos {controlled-load | guaranteed-delay} [audio | video]
5. req-qos {controlled-load | guaranteed-delay} [audio | video] [bandwidth [default bandwidth-value] [max bandwidth-value]]
6. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 enable</td>
<td>Enables privileged EXEC mode. • Enter your password if prompted.</td>
</tr>
<tr>
<td>Example: Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td>Step 2 configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 3 dial-peer voice tag voip</td>
<td>Enters dial peer voice configuration mode.</td>
</tr>
<tr>
<td>Example: Router(config)# dial-peer 77 voip</td>
<td></td>
</tr>
<tr>
<td>Step 4 no acc-qos {controlled-load</td>
<td>guaranteed-delay} [audio</td>
</tr>
<tr>
<td>Example: Router(config-dial-peer)# no acc-qos controlled-load</td>
<td></td>
</tr>
<tr>
<td>Step 5 req-qos {controlled-load</td>
<td>guaranteed-delay} [audio</td>
</tr>
</tbody>
</table>
Configuring EO to EO DO to DO and DO to EO at the Dial Peer

Perform this task to configure support for EO to EO, DO to DO, and DO to EO at the dial peer level.

SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `dial-peer voice tag voip`
4. `no acc-qos {controlled-load | guaranteed-delay} [audio | video]`
5. `req-qos {controlled-load | guaranteed-delay} [audio | video] [bandwidth [default bandwidth-value] [max bandwidth-value]]`
6. `exit`
7. `interface type slot/port`
8. `ip rsvp bandwidth [reservable-bw [max-reservable-bw] [sub-pool reservable-bw]]`
9. `exit`

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router&gt; enable</td>
<td>Enter your password if prompted.</td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
</tr>
<tr>
<td>------</td>
<td>------------------</td>
</tr>
<tr>
<td>Step 2</td>
<td>configure terminal</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>Router# configure terminal</td>
</tr>
<tr>
<td>Step 3</td>
<td>dial-peer voice tag voip</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>Router(config)# dial-peer voice 77 voip</td>
</tr>
<tr>
<td>Step 4</td>
<td>no acc-qos {controlled-load</td>
</tr>
<tr>
<td></td>
<td>guaranteed-delay} [audio</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>Router(config-dial-peer)# no acc-qos controlled-load</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td>req-qos {controlled-load</td>
</tr>
<tr>
<td></td>
<td>guaranteed-delay} [audio</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>Router(config-dial-peer)# req-qos controlled-load</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Note: Configure the req-qos command using the same keyword that you used to configure the acc-qos command, either controlled-load or guaranteed-delay. That is, if you configured the acc-qos controlled-load command in the previous step, then use the req-qos controlled-load command here.</td>
</tr>
<tr>
<td>Step 6</td>
<td>exit</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>Router(config-dial-peer)# exit</td>
</tr>
<tr>
<td>Step 7</td>
<td>interface type slot/port</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>Router(config)# interface FastEthernet 0/1</td>
</tr>
</tbody>
</table>
### Configuring Mandatory RSVP on the Dial Peer

Perform this task to configure Mandatory RSVP on the dial peer. This configuration ensures that the call does not connect if sufficient bandwidth is not allocated.

**SUMMARY STEPS**

1. `enable`
2. `configure terminal`
3. `dial-peer voice tag voip`
4. `acc-qos {best-effort | controlled-load | guaranteed-delay} [audio | video]`
5. `req-qos {best-effort [audio | video] | {controlled-load | guaranteed-delay} [audio | video] [bandwidth [default bandwidth-value] [max bandwidth-value]]}`
6. `end`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
</tbody>
</table>
### Configuring Midcall RSVP Failure Policies

Perform this task to enable call handling policies for a midcall RSVP failure.

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 3** | **Command or Action:** dial-peer voice *tag* voip  
**Example:** Router(config)# dial-peer 77 voip | Enters dial peer voice configuration mode. |
| **Step 4** | **Command or Action:** acc-qos {best-effort | controlled-load | guaranteed-delay} [audio | video]  
**Example:** Router(config-dial-peer)# acc-qos best-effort | Configures mandatory RSVP on the dial-peer.  
- **best-effort** --Indicates that Resource Reservation Protocol (RSVP) makes no bandwidth reservation. This is the default.  
- **controlled-load** --Indicates that RSVP guarantees a single level of preferential service, presumed to correlate to a delay boundary. The controlled load service uses admission (or capacity) control to ensure that preferential service is received even when the bandwidth is overloaded.  
- **guaranteed-delay** --Indicates that RSVP reserves bandwidth and guarantees a minimum bit rate and preferential queueing if the bandwidth reserved is not exceeded. |
| **Step 5** | **Command or Action:** req-qos {best-effort [audio | video] | controlled-load | guaranteed-delay} [audio | video] [bandwidth [default bandwidth-value] [max bandwidth-value]]  
**Example:** Router(config-dial-peer)# req-qos controlled-load | Configures mandatory RSVP on the dial-peer.  
- Calls continue even if there is a drop in the bandwidth reservation. |
| **Step 6** | **Command or Action:** end  
**Example:** Router(config-dial-peer)# end | (Optional) Exits dial peer voice configuration mode and returns to privileged EXEC mode. |
**SUMMARY STEPS**

1. `enable`
2. `configure terminal`
3. `dial-peer voice tag voip`
4. `voice-class sip rsvp-fail-policy {video | voice} post-alert {optional keep-alive | mandatory {keep-alive | disconnect retry retry-attempts}} interval seconds`
5. `end`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router&gt; enable</td>
</tr>
<tr>
<td>• Enter your password if prompted.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router# configure terminal</td>
</tr>
<tr>
<td><strong>Step 3</strong> dial-peer voice tag voip</td>
<td>Enters dial peer voice configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config)# dial-peer voice 66 voip</td>
</tr>
<tr>
<td><strong>Step 4</strong> voice-class sip rsvp-fail-policy {video</td>
<td>voice} post-alert {optional keep-alive</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-dial-peer)# voice-class sip rsvp-fail-policy voice post-alert mandatory keep-alive interval 50</td>
</tr>
<tr>
<td>• optional keep-alive -- The keepalive messages are sent when RSVP fails only if RSVP negotiation is optional.</td>
<td></td>
</tr>
<tr>
<td>• mandatory keep-alive -- The keepalive messages are sent when RSVP fails only if RSVP negotiation is mandatory.</td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong> Keepalive messages are sent at 30-second intervals when a postalert call fails to negotiate RSVP regardless of the RSVP negotiation setting (mandatory or optional).</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> end</td>
<td>Exits dial peer voice configuration mode and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config-dial-peer)# end</td>
</tr>
</tbody>
</table>
## Configuring DSCP Values

Perform this task to configure different Differentiated Services Code Point (DSCP) values based on RSVP status.

### SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `dial-peer voice tag voip`
4. `ip qos dscp {dscp-value | set-af | set-Cs | default | ef} {signaling | media [rsvp-pass | rsvp-fail] | video[rsvp-none | rsvp-pass | rsvp-fail]}`
5. `end`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>enable</code></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Router&gt; enable</td>
</tr>
<tr>
<td>Step 2</td>
<td><code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>Router# configure terminal</td>
</tr>
<tr>
<td>Step 3</td>
<td><code>dial-peer voice tag voip</code></td>
<td>Enters dial peer voice configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>Router(config)# dial-peer voice 66 voip</td>
</tr>
<tr>
<td>Step 4</td>
<td>`ip qos dscp {dscp-value</td>
<td>set-af</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <em>media rsvp-pass</em> -- Specifies that the DSCP value applies to media packets with successful RSVP reservations.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>Router(config-dial-peer)# ip qos dscp af11 media rsvp-pass</td>
</tr>
</tbody>
</table>
### Configuring Support for Interworking Between RSVP Capable and RSVP Incapable Networks

**Command or Action** | **Purpose** |
--- | --- |
• media rsvp-fail | Specifies that the DSCP value applies to packets (media or video) with failed RSVP reservations.  
• The default DSCP value for all media (voice and fax) packets is ef.  
--- | --- |
**Note** | You must configure the DSCP values for all cases: media rsvp-pass and media rsvp-fail. |
**Step 5** | end |
**Example:** | Exits dial peer voice configuration mode and returns to privileged EXEC mode. |
Router(config-dial-peer)# end |

---

## Configuring an Application ID

Perform this task to configure a specific application ID for RSVP establishment.

### SUMMARY STEPS

1. enable  
2. configure terminal  
3. dial-peer voice tag voip  
4. ip qos policy-locator {video | voice} [app app-string] [guid guid-string] [sapp subapp-string] [ver version-string]  
5. end

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
**Step 1** | enable | Enables privileged EXEC mode.  
**Example:**  
Router> enable |
**Step 2** | configure terminal | Enters global configuration mode.  
**Example:**  
Router# configure terminal |
### Purpose

Command or Action | Purpose
--- | ---
**Step 3** | **dial-peer voice tag voip**
Example: Router(config)# dial-peer voice 66 voip | Enters dial peer voice configuration mode.

**Step 4** | **ip qos policy-locator {video | voice} [app app-string] [guid guid-string] [sapp subapp-string] [ver version-string]**
Example: Router(config-dial-peer)# ip qos policy-locator voice | Configures a QoS policy locator (application ID) used to deploy RSVP policies for specifying bandwidth reservations on Cisco IOS Session Initiation Protocol (SIP) devices.

**Step 5** | **end**
Example: Router(config-dial-peer)# end | Exits dial peer voice configuration mode and returns to privileged EXEC mode.

---

### Configuring Priority

Perform this task to configure priorities for call preemption.

#### SUMMARY STEPS

1. enable
2. configure terminal
3. dial-peer voice *tag* voip
4. ip qos defending-priority *defending-pri-value*
5. ip qos preemption-priority *preemption-pri-value*
6. end

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>enable</strong></td>
</tr>
</tbody>
</table>
Example: | Enables privileged EXEC mode.  
- Enter your password if prompted. |
### Command or Action

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Command or Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enters global configuration mode.</td>
<td><code>Router&gt; enable</code></td>
</tr>
<tr>
<td>Enters dial peer voice configuration mode.</td>
<td><code>configure terminal</code></td>
</tr>
<tr>
<td>Enters dial peer voice configuration mode.</td>
<td><code>dial-peer voice tag voip</code></td>
</tr>
<tr>
<td>Configures the RSVP defending priority value for determining QoS.</td>
<td><code>ip qos defending-priority defending-pri-value</code></td>
</tr>
<tr>
<td>Configures the RSVP preemption priority value for determining QoS.</td>
<td><code>ip qos preemption-priority preemption-pri-value</code></td>
</tr>
<tr>
<td>Exits dial peer configuration mode and returns to privileged EXEC mode.</td>
<td><code>end</code></td>
</tr>
</tbody>
</table>

### Troubleshooting the Support for Interworking Between RSVP Capable and RSVP Incapable Networks Feature

Use the following commands to debug any errors that you may encounter when you configure the Support for Interworking Between RSVP Capable and RSVP Incapable Networks feature:

- `debug call rsvp-sync events`
- `debug call rsvp-sync func-trace`
- `debug ccsip all`
Verifying Support for Interworking Between RSVP Capable and RSVP Incapable Networks

This task explains how to display information to verify the configuration for the Support for Interworking Between RSVP Capable and RSVP Incapable Networks feature. These commands need not be entered in any specific order.

SUMMARY STEPS

1. enable
2. show sip-ua calls
3. show ip rsvp installed
4. show ip rsvp reservation
5. show ip rsvp interface detail [interface-type number]
6. show sccp connections details
7. show sccp connections rsvp
8. show sccp connections internal
9. show sccp [all | connections | statistics]

DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router&gt; enable</td>
<td>- Enter your password if prompted.</td>
</tr>
<tr>
<td>Step 2</td>
<td>show sip-ua calls</td>
<td>(Optional) Displays active user agent client (UAC) and user agent server (UAS) information on SIP calls.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router# show sip-ua calls</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>show ip rsvp installed</td>
<td>(Optional) Displays RSVP-related installed filters and corresponding bandwidth information.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router# show ip rsvp installed</td>
<td></td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------</td>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td>4</td>
<td>show ip rsvp reservation</td>
<td>(Optional) Displays RSVP-related receiver information currently in the database.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router# show ip rsvp reservation</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>show ip rsvp interface detail [interface-type number]</td>
<td>(Optional) Displays the interface configuration for hello.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router# show ip rsvp interface detail GigabitEthernet 0/0</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>show sccp connections details</td>
<td>(Optional) Displays SCCP connection details, such as call-leg details.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router# show sccp connections details</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>show sccp connections rsvp</td>
<td>(Optional) Displays information about active SCCP connections that are using RSVP.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router# show sccp connections rsvp</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>show sccp connections internal</td>
<td>(Optional) Displays the internal SCCP details, such as time-stamp values.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router# show sccp connections internal</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>show sccp [all</td>
<td>connections</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router# show sccp statistics</td>
<td></td>
</tr>
</tbody>
</table>

**Feature Information for Configuring Support for Interworking Between RSVP Capable and RSVP Incapable Networks**

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 . An account on Cisco.com is not required.

ISR Feature table entry
### Table 27: Feature Information for Interworking Between RSVP Capable and RSVP Incapable Networks

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interworking Between RSVP Capable and RSVP Incapable Networks</td>
<td>15.0(1)XA 15.1(3)T</td>
<td>The Support for Interworking Between RSVP Capable and RSVP Incapable Networks feature provides precondition-based RSVP support for basic audio call and supplementary services on the Cisco UBE. Support forConfiguring EO-EO, DO-DO andDO-EO support on dial peer was introduced in 15.1(3)T.15.1(3)T--Configuring EO-EO, DO-DO and DO-EO support on dial peer.</td>
</tr>
</tbody>
</table>

ASR Feature table entry

### Table 28: Feature Information for Interworking Between RSVP Capable and RSVP Incapable Networks

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interworking Between RSVP Capable and RSVP Incapable Networks</td>
<td>TBD</td>
<td>The Support for Interworking Between RSVP Capable and RSVP Incapable Networks feature provides precondition-based RSVP support for basic audio call and supplementary services on the Cisco UBE.</td>
</tr>
</tbody>
</table>
VoIP for IPv6

VoIPv6 adds IPv6 capability to existing VoIP features. VoIPv6 requires IPv6 and IPv4 dual-stack support on voice gateways and MTP, IPv6 support for SIP trunks, and SCCP-controlled analog voice phones. In addition, the SBC functionality of connecting SIP IPv4 or H.323 IPv4 network to SIP IPv6 network is implemented on a Cisco Unified Border Element to facilitate migration from VoIPv4 to VoIPv6.

- Prerequisites, page 273
- Configuring VoIP for IPv6, page 273
- Feature Information for VoIP for IPv6, page 274

Prerequisites

Listing the minimum SW release is required. Use the following wording:

Cisco Unified Border Element

- Cisco IOS Release 12.4(22)T or a later release must be installed and running on your Cisco Unified Border Element.

Cisco Unified Border Element (Enterprise)

- Cisco IOS XE Release 3.3S or a later release must be installed and running on your Cisco ASR 1000 Series Router.

Configuring VoIP for IPv6

To enable this feature, see the "Implementing VoIP for IPv6" section in the Cisco IOS IPv6 Configuration Guide, Release 15.0.

Detailed command information for the VoIP for IPv6 commands is located in the Cisco IOS IPv6 Command Reference.
Feature Information for VoIP for IPv6

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 . An account on Cisco.com is not required.

Table 29: Feature Information for VoIP for IPv6

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>VoIP for IPv6</td>
<td>12.4(22)T</td>
<td>VoIPv6 adds IPv6 capability to existing VoIP features. Additionally, the SBC functionality of connecting SIP IPv4 or H.323 IPv4 network to SIP IPv6 network is implemented on a Cisco Unified Border Element to facilitate migration from VoIPv4 to VoIPv6. The following commands were introduced or modified: None.</td>
</tr>
</tbody>
</table>

Table 30: Feature Information for VoIP for IPv6

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>VoIP for IPv6</td>
<td>Cisco IOS XE Release 3.3S</td>
<td>VoIPv6 adds IPv6 capability to existing VoIP features. Additionally, the SBC functionality of connecting SIP IPv4 or H.323 IPv4 network to SIP IPv6 network is implemented on a Cisco Unified Border Element to facilitate migration from VoIPv4 to VoIPv6. The following commands were introduced or modified: None.</td>
</tr>
</tbody>
</table>
Mid-call Signaling Consumption

The Cisco Unified Border Element BE Mid-call Signaling support aims to reduce the interoperability issues that arise due to consuming mid-call RE-INVITES/UPDATES.

Mid-call Re-INVITEs/UPDATEs can be consumed in the following ways:

- Mid-call Signaling Passthrough - Media Change
- Mid-call Signaling Block
- Mid-call Signaling Codec Preservation

**Note**

This feature should be used as a last resort only when there is no other option in CUBE. This is because configuring this feature can break video-related features. For Delay-offer Re-INVITE, the configured codec will be passed as an offer in 200 message to change the codec, the transcoder is added in the answer.

- Feature Information for Mid-call Signaling, page 275
- Prerequisites, page 276
- Mid-call Signaling Passthrough - Media Change, page 277
- Mid-call Signaling Block, page 280
- Mid Call Codec Preservation, page 282

Feature Information for Mid-call Signaling

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 . An account on Cisco.com is not required.
### Table 31: Feature Information for Mid-call Signaling

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mid-call Re-INVITE Consumption</td>
<td>Cisco IOS 15.2(1)T</td>
<td>The Mid-call Re-INVITE consumption feature consumes mid-call Re-INVITEs from CUBE and helps to avoid interoperability issues because of these re-invites. The following commands were introduced or modified: midcall-signaling.</td>
</tr>
<tr>
<td></td>
<td>Cisco IOS XE 3.6S</td>
<td></td>
</tr>
<tr>
<td>Mid-call Codec Preservation</td>
<td>Cisco IOS 15.3(2)T</td>
<td>The Mid-call Codec Preservation feature helps to disable codec negotiation in the middle of a call and preserves the codec negotiated before the call. The following commands were introduced or modified: midcall-signaling preserve-codec, voice-class sip midcall-signaling preserve-codec.</td>
</tr>
<tr>
<td></td>
<td>Cisco IOS XE 3.9S</td>
<td></td>
</tr>
<tr>
<td>Mid-call Re-INVITE Consumption</td>
<td>Cisco IOS 15.5(3)M</td>
<td>Mid-call signaling Re-INVITE consumption is enhanced to support:</td>
</tr>
<tr>
<td></td>
<td>Cisco IOS XE 3.16S</td>
<td>- Re-INVITE based call transfer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Call transfer with REFER Consume</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Normalization of call hold in a call set-up</td>
</tr>
</tbody>
</table>

### Prerequisites

- Enable CUBE application on a device
- Cisco IOS Release 15.2(1)T or later, or Cisco IOS-XE Release 15.2(2)S or later must be installed.
- **supplementary-service media-renegotiate** must be configured in global voice service voip mode.
Mid-call Signaling Passthrough - Media Change

Passthrough media change method optimizes or consumes mid-call, media-related signaling within the call. Mid-call signaling changes will be passed through only when bidirectional media like T.38 or video is added. The command `midcall-signaling passthru media-change` needs to be configured to enable passthrough media change.

Restrictions for Mid-call Signaling Passthrough - Media Change

- SIP-H.323 calls are not supported.
- TDM Gateways are not supported.
- Session Description Protocol (SDP) -passthrough is not supported
- When `codec` T is configured, the offer from CUBE has only audio codecs, and so the video codecs are not consumed.
- Re-invites are not consumed if media flow-around is configured.
- Re-invites are not consumed if media anti-tromboning is configured.
- Video transcoding is not supported.
- Secure Real-time Protocol - Real-time Protocol (SRTP-RTP) supplementary services are not supported.
- Multicast Music On Hold (MMOH) is not supported.
- When the `midcall-signaling passthru media-change` command is configured and high-density transcoder is enabled, there might be some impact on Digital Signal Processing (DSP) resources as the transcoder might be used for all the calls.
- Session timer is handled leg by leg whenever this feature is configured.
- More than two m-lines in the SDP is not supported.
- Alternative Network Address Types (ANAT) is not supported.
- Video calls and Application streams are not supported when mid-call signaling block is configured.

Behavior of Mid-call Re-INVITE Consumption

- If mid-call signaling block is enabled on either of call-legs, video parameters and application streams are not negotiated, and are rejected in the answer.
- When flow around and offer-all is configured, CUBE performs codec renegotiation even if mid-call signaling block is configured globally.
- Below behavior is for refer consume scenario:
  - REFER consume is supported for blind, alert and consult call transfers.
  - Existing codecs or DTMF is used for local bridging of new call legs. No Re-INVITE or UPDATE is sent for media re-negotiation after REFER.
• Call gets dropped when DSP is required but not available.
• A call can be escalated to video only if transferee and transfer-to dial-peers do not have mid-call signaling block configured.
• Video calls are de-escalated if mid-call signaling block configuration on transfer-to dial-peer.
• For Re-INVITE based call-transfer involving Cisco Unified Communications Manager, all Re-INVITE are locally answered and transcoder is invoked if negotiated codecs are different than the codecs before call-transfer.

• The following table provides the details of the behavior when the initial call is establish without 'sendrecv' parameter, that means, the initial call is established with 'sendonly', 'recvonly' or 'inactive'.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>If an Offer is received with 'sendonly' and mid-call block is configured on any or both call legs</td>
<td>Offer is sent with 'sendrecv'.</td>
</tr>
<tr>
<td>If an Answer is received with 'sendonly' and the peer leg supports mid-call signaling</td>
<td>Answer is sent with 'sendonly'. Resume transaction is end-to-end.</td>
</tr>
<tr>
<td>If an Answer is received with 'sendonly' and the peer leg does not supports mid-call signaling</td>
<td>Answer is sent with 'sendrecv'. Resume transaction is consumed.</td>
</tr>
<tr>
<td>If Offer as well as Answer is received with 'sendonly' and Offering leg does not support mid-call signaling</td>
<td>Answer is sent with 'recvonly'. Resume from Offering leg is end-to-end. Resume from answering leg is consumed.</td>
</tr>
<tr>
<td>If Offer as well as Answer is received with 'sendonly' and Answering leg does not support mid-call signaling</td>
<td>Answer is sent with 'inactive'. Resume from Offering leg is consumed. Resume from answering leg is end-to-end.</td>
</tr>
<tr>
<td>If Offer as well as Answer is received with 'sendonly' and both legs do not support mid-call signaling</td>
<td>Answer is sent with 'recvonly'. Resume transaction is consumed.</td>
</tr>
</tbody>
</table>

**Configuring Passthrough of Mid-call Signalling**

Perform this task to configure passthrough of mid-call signaling (as Re-invites) only when bidirectional media is added.
SUMMARY STEPS

1. enable
2. configure terminal
3. Configure passthrough of mid-call signaling changes only when bidirectional media is added.
   - In Global VoIP SIP configuration mode
     midcall-signaling passthru media-change
   - In dial-peer configuration mode
     voice-class sip mid-call signaling passthru media-change
4. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device&gt; enable</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Step 2 configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 3 Configure passthrough of mid-call signaling changes only when bidirectional media is added.</td>
<td>Re-Invites are passed through only when bidirectional media is added.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>In Global VoIP SIP configuration mode:</td>
<td></td>
</tr>
<tr>
<td>Device(config)# voice service voip</td>
<td></td>
</tr>
<tr>
<td>Device(config)# sip</td>
<td></td>
</tr>
<tr>
<td>Device(conf-serv-sip)# midcall-signaling passthru media-change</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>In Dial-peer configuration mode:</td>
<td></td>
</tr>
<tr>
<td>Device(config)# dial-peer voice 2 voip</td>
<td></td>
</tr>
<tr>
<td>Device(config-dial-peer)# voice-class sip mid-call signaling passthru media-change</td>
<td></td>
</tr>
</tbody>
</table>
Example Configuring Passthrough SIP Messages at Dial Peer Level

The following example shows how to passthrough SIP messages at the dial peer level:

dial-peer voice 600 voip
  destination-pattern 2222222222
  session protocol sipv2
  session target ipv4:9.45.38.39:9001
  voice-class sip mid-call signaling passthru media-change
  incoming called-number 1111111111
  voice-class codec 2 offer-all

dial-peer voice 400 voip
  destination-pattern 1111111111
  session protocol sipv2
  session target ipv4:9.45.38.39:9000
  incoming called-number 2222222222
  voice-class codec 1 offer-all

Example Configuring Passthrough SIP Messages at the Global Level

The following example shows how to passthrough SIP messages at the global level:

Device(config)# voice service voip
Device(conf-voi-serv)# no ip address trusted authenticate
Device(conf-voi-serv)# allow-connections sip to sip
Device(conf-voi-serv)# sip
Device(conf-serv-sip)# midcall-signaling passthru media-change

Mid-call Signaling Block

The Block method blocks all mid-call media-related signaling to the specific SIP trunk. The command `midcall-signaling block` needs to be configured to enable this behavior. Video escalation and T.38 call flow are rejected when the `midcall-signaling block` command is configured. This command should be configured only when basic call is the focus and mid-call can be consumed.

Restrictions for Mid-Call Signaling Block

- SIP-H.323 calls are not supported.
- TDM Gateways are not supported.
- Session Description Protocol (SDP) -passthrough is not supported
- Video calls and Application streams are not supported.
- When media flow-around is configured, Mid-call INVITE is rejected with 488 error message.
- Re-invites are not consumed if media anti-tromboning is configured.
• SRTP-RTP supplementary services are not supported.
• Multicast Music On Hold (MMOH) is not supported.
• When the `midcall-signaling passthru media-change` command is configured and high-density transcoder is enabled, there might be some impact on Digital Signal Processing (DSP) resources as the transcoder might be used for all the calls.
• Session timer is handled leg by leg whenever this feature is configured.
• More than two m-lines in the SDP is not supported.
• Alternative Network Address Types (ANAT) is not supported.

Blocking Mid-Call Signaling

Perform this task to block mid-call signaling:

**SUMMARY STEPS**

1. `enable`
2. `configure terminal`
3. Configure blocking of mid-call signaling changes:
   - In Global VoIP SIP configuration mode
     `midcall-signaling block`
   - In dial-peer configuration mode
     `voice-class sip mid-call signaling block`
4. `end`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>enable</code></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> Device&gt; <code>enable</code></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Step 2</td>
<td><code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> Device# <code>configure terminal</code></td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>Configure blocking of mid-call signaling changes:</td>
<td>Mid-call signaling is always blocked.</td>
</tr>
<tr>
<td></td>
<td>• In Global VoIP SIP configuration mode</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>midcall-signaling block</code></td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

- In dial-peer configuration mode
  
  `voice-class sip mid-call signaling block`

### Purpose

**Example:**

In Global VoIP SIP configuration mode:

```bash
Device(config)# voice service voip
Device(conf-voi-serv)# sip
Device(conf-serv-sip)# midcall-signaling block
```

**Example:**

In Dial-peer configuration mode:

```bash
Device(config)# dial-peer voice 2 voip
Device(config-dial-peer)# voice-class sip mid-call signaling block
```

### Step 4

<table>
<thead>
<tr>
<th>Step 4</th>
<th>end</th>
<th>Exits to privileged EXEC mode.</th>
</tr>
</thead>
</table>

### Example Blocking SIP Messages at Dial Peer Level

```bash
dial-peer voice 107 voip
  destination-pattern 74000
  session protocol sipv2
  session target ipv4:9.45.36.9
  incoming called-number 84000
  voice-class codec 1 offer-all
!
dial-peer voice 110 voip
  destination-pattern 84000
  session protocol sipv2
  session target ipv4:9.45.35.2
  incoming called-number 74000
  voice-class codec 1 offer-all
  voice-class sip mid-call signaling block
```

### Example: Blocking SIP Messages at the Global Level

The following example shows how to block SIP messages at the global Level

```bash
Device(config)# voice service voip
Device(config-voi-serv)# no ip address trusted authenticate
Device(config-voi-serv)# allow-connections sip to sip
Device(config-voi-serv)# sip
Device(config-serv-sip)# midcall-signaling block
```

### Mid Call Codec Preservation

Mid call codec preservation defines whether a codec can be negotiated after a call has been initiated. You can enable or disable codec negotiation in the middle of a call.
Configuring Mid Call Codec Preservation

This task disables codec negotiation in the middle of a call and preserves the codec negotiated before the call.

SUMMARY STEPS

1. enable
2. configure terminal
3. Enter one of the following to disable midcall codec renegotiation:
   • In Global VoIP SIP configuration mode
     midcall-signaling preserve-codec
   • In dial-peer configuration mode
     voice-class sip midcall-signaling preserve-codec
4. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device&gt; enable</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> Enter one of the following to disable midcall codec renegotiation:</td>
<td>Disables codec negotiation in the middle of a call and preserves the codec negotiated before the call.</td>
</tr>
<tr>
<td>• In Global VoIP SIP configuration mode</td>
<td></td>
</tr>
<tr>
<td>midcall-signaling preserve-codec</td>
<td></td>
</tr>
<tr>
<td>• In dial-peer configuration mode</td>
<td></td>
</tr>
<tr>
<td>voice-class sip midcall-signaling preserve-codec</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config)# voice service voip</td>
<td></td>
</tr>
<tr>
<td>Device(config-voice-serv)# sip</td>
<td></td>
</tr>
<tr>
<td>Device(config-serv-sip)# midcall-signaling preserve-codec</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device(config)# dial-peer voice 10 voip</td>
<td></td>
</tr>
<tr>
<td>Device(conf-dial-peer)# voice-class sip midcall-signaling preserve-codec</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td></td>
</tr>
<tr>
<td>end</td>
<td>Exits to privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device(conf-serv-sip)# end</td>
<td></td>
</tr>
</tbody>
</table>

**Example: Configuring Mid Call Codec Preservation at the Dial Peer Level**

```plaintext
dial-peer voice 107 voip
destination-pattern 74000
session protocol sipv2
session target ipv4:9.45.36.9
incoming called-number 84000
voice-class codec 1 offer-all

dial-peer voice 110 voip
destination-pattern 84000
session protocol sipv2
session target ipv4:9.45.35.2
incoming called-number 74000
voice-class codec 1 offer-all
voice-class sip midcall-signaling preserve-codec
```

**Example: Configuring Mid Call Codec Preservation at the Global Level**

```plaintext
Device(config)# voice service voip
Device(conf-voi-serv)# no ip address trusted authenticate
Device(conf-voi-serv)# allow-connections sip to sip
Device(conf-voi-serv)# sip
Device(conf-serv-sip)# midcall-signaling preserve-codec
```
Support for Software Media Termination Point

The Support for Software Media Termination Point (MTP) feature bridges the media streams between two connections allowing Cisco Unified Communications Manager (Cisco UCM) to relay calls that are routed through SIP or H.323 endpoints via Skinny Call Control Protocol (SCCP) commands. These commands allow Cisco UCM to establish an MTP for call signaling.

- Finding Feature Information, page 285
- Information About Support for Software Media Termination Point, page 285
- How to Configure Support for Software Media Termination Point, page 286
- Prerequisites, page 286
- Restrictions, page 286
- Configuring Support for Software Media Termination Point, page 286
- Feature Information for Support for Software Media Termination Point, page 291

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About Support for Software Media Termination Point

This feature extends the software MTP support to the Cisco Unified Border Element (Enterprise). Software MTP is an essential component of large-scale deployments of Cisco UCM. This feature enables new capabilities...
so that the Cisco UBE can function as an Enterprise Edge Cisco Session Border Controller for large-scale deployments that are moving to SIP trunking.

How to Configure Support for Software Media Termination Point

Prerequisites

• For the software MTP to function properly, codec and packetization must be configured the same way on both in call legs and out call legs.

Cisco Unified Border Element (Enterprise)

• Cisco IOS XE Release 2.6 or a later release must be installed and running on your Cisco ASR 1000 Series Router.

Restrictions

• RSVP Agent is not supported in software MTP.
• Hardware MTP for repacketization is not supported.
• Call Threshold is not supported for standalone software MTP.
• Per-call debugging is not supported.

Configuring Support for Software Media Termination Point

To enable and configure the Support for Software Media Termination Point feature, perform the following task.
SUMMARY STEPS

1. enable
2. configure terminal
3. sccp local interface-type interface-number [port port-number]
4. sccp ccm {ipv4-address | ipv6-address | dns} identifier identifier-number [port port-number] version version-number
5. sccp
6. sccp ccm group group-number
7. associate ccm identifier-number priority number
8. associate profile profile-identifier register device-name
9. dspfarm profile profile-identifier {conference | mtp | transcode} [security]
10. maximum sessions {hardware | software} number
11. associate application sccp
12. no shutdown

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> sccp local interface-type interface-number [port port-number]</td>
<td>Selects the local interface that SCCP applications (transcoding and</td>
</tr>
<tr>
<td>Example:</td>
<td>conferencing) use to register with Cisco UCM.</td>
</tr>
<tr>
<td>Router(config)# sccp local</td>
<td>• interface type --Can be an interface address or a virtual-interface</td>
</tr>
<tr>
<td>gigabitethernet0/0/0</td>
<td>address such as Ethernet.</td>
</tr>
<tr>
<td></td>
<td>• interface number --Interface number that the SCCP application uses</td>
</tr>
<tr>
<td></td>
<td>to register with Cisco UCM.</td>
</tr>
<tr>
<td></td>
<td>• (Optional) port port-number--Port number used by the selected</td>
</tr>
<tr>
<td></td>
<td>interface. Range is 1025 to 65535. Default is 2000.</td>
</tr>
<tr>
<td><strong>Step 4</strong> sccp ccm {ipv4-address</td>
<td>Adds a Cisco UCM server to the list of available servers and sets the</td>
</tr>
<tr>
<td>ipv6-address</td>
<td>dns} identifier identifier-number [port</td>
</tr>
<tr>
<td>port-number] version version-number</td>
<td>• ipv4-address --IP version 4 address of the Cisco UCM server.</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
</tbody>
</table>
| Router(config)# sccp ccm 10.1.1.1 identifier 1 version 7.0+ | • *ipv6-address* -- IP version 6 address of the Cisco UCM server.  
• *dns* -- DNS name.  
• *identifier* -- Specifies the number that identifies the Cisco UCM server. Range is 1 to 65535.  
• *port* *port-number* (Optional) -- Specifies the TCP port number. Range is 1025 to 65535. Default is 2000.  
• *version* *version-number* -- Cisco UCM version. Valid versions are 3.0, 3.1, 3.2, 3.3, 4.0, 4.1, 5.0.1, 6.0, and 7.0+. There is no default value. |
| **Step 5**        |         |
| sccp              | Enables the Skinny Client Control Protocol (SCCP) and its related applications (transcoding and conferencing). |
| **Example:**      |         |
| Router(config)# sccp |         |
| **Step 6**        |         |
| sccp ccm group *group-number* | Creates a Cisco UCM group and enters SCCP Cisco UCM configuration mode.  
• *group-number* -- Identifies the Cisco UCM group. Range is 1 to 50. |
| **Example:**      |         |
| Router(config)# sccp ccm group 10 |         |
| **Step 7**        |         |
| associate ccm *identifier-number* *priority number* | Associates a Cisco UCM with a Cisco UCM group and establishes its priority within the group:  
• *identifier-number* -- Identifies the Cisco UCM. Range is 1 to 65535. There is no default value.  
• *priority number* -- Priority of the Cisco UCM within the Cisco UCM group. Range is 1 to 4. There is no default value. The highest priority is 1. |
| **Example:**      |         |
| Router(config-sccp-ccm)# associate ccm 10 priority 3 |         |
| **Step 8**        |         |
| associate profile *profile-identifier* *register device-name* | Associates a DSP farm profile with a Cisco UCM group:  
• *profile-identifier* -- Identifies the DSP farm profile. Range is 1 to 65535. There is no default value.  
• *register device-name* -- Device name in Cisco UCM. A maximum of 15 characters can be entered for the device name. |
| **Example:**      |         |
| Router(config-sccp-ccm)# associate profile 1 register MTP0011 |         |
| **Step 9**        |         |
| dspfarm profile *profile-identifier* \{conference | mtp | transcode | security\} | Enters DSP farm profile configuration mode and defines a profile for DSP farm services:  
• *profile-identifier* -- Number that uniquely identifies a profile. Range is 1 to 65535. There is no default.  
• *conference* -- Enables a profile for conferencing.  
• *mtp* -- Enables a profile for MTP. |
<p>| <strong>Example:</strong>      |         |
| Router(config-sccp-ccm)# dspfarm profile 1 mtp |         |</p>
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>• transcode -- Enables a profile for transcoding.</td>
<td></td>
</tr>
<tr>
<td>• security (Optional) -- Enables a profile for secure DSP farm services.</td>
<td></td>
</tr>
</tbody>
</table>

**Step 10**

maximum sessions \{hardware | software\} number

**Example:**
Router(config-dspfarm-profile)#
maximum sessions software 10

Specifies the maximum number of sessions that are supported by the profile.

- **hardware** -- Number of sessions that MTP hardware resources can support.
- **software** -- Number of sessions that MTP software resources can support.
- **number** -- Number of sessions that are supported by the profile. Range is 0 to x. Default is 0. The x value is determined at run time depending on the number of resources available with the resource provider.

**Step 11**

associate application sccp

**Example:**
Router(config-dspfarm-profile)#
associate application sccp

Associates SCCP to the DSP farm profile.

**Step 12**

no shutdown

**Example:**
Router(config-dspfarm-profile)#
no shutdown

Changes the status of the interface to the UP state.

**Examples**

The following example shows a sample configuration for the Support for Software Media Termination Point feature:

```bash
sccp local GigabitEthernet0/0/1
sccp ccm 10.13.40.148 identifier 1 version 6.0
sccp
!
sccp ccm group 1
bind interface GigabitEthernet0/0/1
associate ccm 1 priority 1
associate profile 6 register RR_RLS6
!
dspfarm profile 6 mtp
codec g711ulaw
maximum sessions software 100
associate application SCCP
!
```

Cisco Unified Border Element Protocol-Independent Features and Setup Configuration Guide, Cisco IOS Release 15M&T
Troubleshooting Tips

To verify and troubleshoot this feature, use the following show commands:

- To verify information about SCCP, use the show sccp command:

  Router# show sccp
  SCCP Admin State: UP
  Gateway IP Address: 10.13.40.157, Port Number: 2000
  IP Precedence: 5
  User Masked Codec list: None
  Call Manager: 10.13.40.148, Port Number: 2000
    Priority: N/A, Version: 6.0, Identifier: 1
    Trustpoint: N/A

- To verify information about the DSPfarm profile, use the show dspfarm profile command:

  Router# show dspfarm profile 6
  Dspfarm Profile Configuration
  Profile ID = 6, Service = MTP, Resource ID = 1
  Profile Description:
  Profile Service Mode : Non Secure
  Profile Admin State : UP
  Profile Operation State : ACTIVE
  Application : SCCP Status : ASSOCIATED
  Resource Provider : NONE Status : NONE
  Number of Resource Configured : 100
  Number of Resource Available : 100
  Hardware Configured Resources : 0
  Hardware Available Resources : 0
  Software Resources : 100
  Codec Configuration
  Codec : g711ulaw, Maximum Packetization Period : 30

- To display statistics for the SCCP connections, use the show sccp connections command:

  Router# show sccp connections
  sess_id conn_id stype mode codec ripaddr rport sport
  16808048 16789079 mtp sendrecv g711u 10.13.40.20 7510 7242
  16808048 16789078 mtp sendrecv g711u 10.13.40.157 6900 18050

- To display information about RTP connections, use the show rtpspi call command:

  Router# show rtpspi call
  RTP Service Provider info:
  No. CallId dstCallId Mode LocalRTP RmtRTP LocalIP RemoteIP SRTP
  22 19 Snd-Rcv 7242 17510 0x90D080F 0x90D0814 0
  19 22 Snd-Rcv 18050 6900 0x90D080F 0x90D080F 0

- To display information about VoIP RTP connections, use the show voip rtp connections command:

  Router# show voip rtp connections
  VoIP RTP Port Usage Information
  Max Ports Available: 30000, Ports Reserved: 100, Ports in Use: 102
Port range not configured, Min: 5500, Max: 65499
VoIP RTP active connections:

<table>
<thead>
<tr>
<th>No.</th>
<th>CallId</th>
<th>dstCallId</th>
<th>LocalRTP</th>
<th>RmtRTP</th>
<th>LocalIP</th>
<th>RemoteIP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>114</td>
<td>117</td>
<td>19822</td>
<td>24556</td>
<td>10.13.40.157</td>
<td>10.13.40.157</td>
</tr>
<tr>
<td>3</td>
<td>116</td>
<td>115</td>
<td>19176</td>
<td>52625</td>
<td>10.13.40.157</td>
<td>10.13.40.20</td>
</tr>
<tr>
<td>4</td>
<td>117</td>
<td>114</td>
<td>16526</td>
<td>52624</td>
<td>10.13.40.157</td>
<td>10.13.40.20</td>
</tr>
</tbody>
</table>

- Additional, more specific, **show** commands that can be used include the following:
  - **show scp connection callid**
  - **show scp connection connid**
  - **show scp connection sessionid**
  - **show rtpspi call callid**
  - **show rtpspi stat callid**
  - **show voip rtp connection callid**
  - **show voip rtp connection type**

- To isolate specific problems, use the **debug scp** command:
  - **debug scp [all | config | errors | events | keepalive | messages | packets | parser | tls]**

**Feature Information for Support for Software Media Termination Point**

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 . An account on Cisco.com is not required.

**Feature History Table for the ASR**

**Table 32: Feature Information for Support for Software Media Termination Point**

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support for Software Media Termination Point</td>
<td>Cisco IOS XE Release 2.6 S</td>
<td>Software Media Termination Point (MTP) provides the capability for Cisco Unified Communications Manager (Cisco UCM) to interact with a voice gateway via Skinny Client Control Protocol (SCCP) commands. These commands allow the Cisco UCM to establish an MTP for call signaling.</td>
</tr>
</tbody>
</table>
Cisco Unified Communication Trusted Firewall Control

Cisco Unified Communications Trusted Firewall Control pushes intelligent services onto the network through a Trusted Relay Point (TRP) firewall. Firewall traversal is accomplished using Session Traversal Utilities for NAT (STUN) on a TRP collocated with a Cisco Unified Communications Manager Express (Cisco Unified CME) or a Cisco Unified Border Element.

- Finding Feature Information, page 293
- Prerequisites, page 293
- Configuring Cisco Unified Communication Trusted Firewall Control, page 294
- Feature Information for Cisco Unified Communication Trusted Firewall Control, page 294

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites

Cisco Unified Border Element

- Cisco IOS Release 12.4(22)T or a later release must be installed and running on your Cisco Unified Border Element.
Cisco Unified Border Element (Enterprise)

- Cisco IOS XE Release 3.3S or a later release must be installed and running on your Cisco ASR 1000 Series Router.

Configuring Cisco Unified Communication Trusted Firewall Control

To enable this feature, see the "Cisco Unified Communications Trusted Firewall Control" feature guide. Detailed command information for the `stun`, `stun flowdata agent-id`, `stun flowdata keepalive`, `stun flowdata shared-secret`, `stun usage firewall-traversal flowdata`, `voice-class stun-usage` commands is located in the Cisco IOS Voice Command Reference.

Feature Information for Cisco Unified Communication Trusted Firewall Control

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 . An account on Cisco.com is not required.

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco Unified Communications Trusted Firewall Control</td>
<td>12.4(22)T</td>
<td>Cisco Unified Communications Trusted Firewall Control pushes intelligent services into the network through Trust Relay Point (TRP). The following commands were introduced or modified: <code>stun</code>, <code>stun flowdata agent-id</code>, <code>stun flowdata keepalive</code>, <code>stun flowdata shared-secret</code>, <code>stun usage firewall-traversal flowdata</code>, <code>voice-class stun-usage</code>.</td>
</tr>
</tbody>
</table>
Table 34: Feature Information for Cisco Unified Communication Trusted Firewall Control

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco Unified Communications Trusted Firewall Control</td>
<td>Cisco IOS XE Release 3.3S</td>
<td>Cisco Unified Communications Trusted Firewall Control pushes intelligent services into the network through Trust Relay Point (TRP). The following commands were introduced or modified: stun, stun flowdata agent-id, stun flowdata keepalive, stun flowdata shared-secret, stun usage firewall-traversal flowdata, voice-class stun-usage.</td>
</tr>
</tbody>
</table>
Cisco Unified Communication Trusted Firewall Control-Version II

Cisco Unified Communications Trusted Firewall Control pushes intelligent services onto the network through a Trusted Relay Point (TRP) firewall. TRP is a Cisco IOS service feature, which is similar to the Resource Reservation Protocol (RSVP) agent. Firewall traversal is accomplished using Session Traversal Utilities for NAT (STUN) on a TRP colocated with a Cisco Unified Communications Manager Express (Cisco Unified CME), Cisco Unified Border Element, and Media Termination Points (MTP).

This release introduces the following features:

- Noncolocated firewall for UC SIP trunks
- Support Firewall traversal for Cisco Unified Border Element call flows in which the media flow through the Media Termination Points such as MTP, Transcoder, or Conference bridge with Trust Relay Point (TRP) enabled.
- Firewall traversal for additional Cisco Unified Border Element call flows using STUN.

- Finding Feature Information, page 297
- Prerequisites for Cisco Unified Communication Trusted Firewall Control-Version II, page 298
- Configuring Cisco Unified Communication Trusted Firewall Control-Version II, page 298
- Feature Information for Cisco Unified Communication Trusted Firewall Control-Version II, page 298

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.
Prerequisites for Cisco Unified Communication Trusted Firewall Control-Version II

Cisco Unified Border Element

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

Cisco Unified Border Element (Enterprise)

- Cisco IOS XE Release 3.3S or a later release must be installed and running on your Cisco ASR 1000 Series Router.

Configuring Cisco Unified Communication Trusted Firewall Control-Version II

To enable this feature, see the "Cisco Unified Communications Trusted Firewall Control-Version II" feature guide.

Detailed command information for the `stun flowdata catlife` command is located in the *Cisco IOS Voice Command Reference*.

Feature Information for Cisco Unified Communication Trusted Firewall Control-Version II

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 . An account on Cisco.com is not required.

**Table 35: Feature Information for Cisco Unified Communication Trusted Firewall Control-Version II**

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco Unified Communication Trusted Firewall Control-Version II</td>
<td>15.0(1)T</td>
<td>Cisco Unified Communications Trusted Firewall Control pushes intelligent services into the network through Trust Relay Point (TRP). The following command was introduced: <code>stun flowdata catlife</code>.</td>
</tr>
</tbody>
</table>
Table 36: Feature Information for Cisco Unified Communication Trusted Firewall Control-Version II

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco Unified Communication Trusted Firewall Control-Version II</td>
<td>Cisco IOS XE Release 3.3S</td>
<td>Cisco Unified Communications Trusted Firewall Control pushes intelligent services into the network through Trust Relay Point (TRP). The following command was introduced: <strong>stun flowdata catlife</strong>.</td>
</tr>
</tbody>
</table>
CHAPTER 33

Domain-Based Routing Support on the Cisco UBE

First Published: June 15, 2011
Last Updated: July 22, 2011

The Domain-based routing feature provides support for matching an outbound dial peer based on the domain name or IP address provided in the request URI of the incoming SIP message or an inbound dial peer.

Domain-based routing enables for calls to be routed on the outbound dialpeer based on the domain name or IP address provided in the request Uniform Resource Identifier (URI) of incoming Session IP message.

- Feature Information for Domain-Based Routing Support on the Cisco UBE, page 301
- Restrictions for Domain-Based Routing Support on the Cisco UBE, page 302
- Information About Domain-Based Routing Support on the Cisco UBE, page 302
- How to Configure Domain-Based Routing Support on the Cisco UBE, page 303
- Configuration Examples for Domain-Based Routing Support on the Cisco UBE, page 308

Feature Information for Domain-Based Routing Support on the Cisco UBE

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 software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.
### Table 37: Feature Information for Domain-Based Routing Support on the Cisco UBE

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain Based Routing Support on the Cisco UBE</td>
<td>15.2(1)T</td>
<td>The domain-based routing enables for calls to be routed on the outbound dial peer based on the domain name or IP address provided in the request URI (Uniform Resource Identifier) of incoming SIP message. The following commands were introduced or modified: <code>call-route</code>, <code>voice-class sip call-route</code>.</td>
</tr>
<tr>
<td>Domain Based Routing Support on the Cisco UBE</td>
<td>Cisco IOS XE Release 3.8S</td>
<td>The domain-based routing enables for calls to be routed on the outbound dial peer based on the domain name or IP address provided in the request URI (Uniform Resource Identifier) of incoming SIP message. The following commands were introduced or modified: <code>call-route</code>, <code>voice-class sip call-route</code>.</td>
</tr>
</tbody>
</table>

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental. © 2011 Cisco Systems, Inc. All rights reserved

### Restrictions for Domain-Based Routing Support on the Cisco UBE

Domain-based routing support is available only for SIP-SIP call flows.

### Information About Domain-Based Routing Support on the Cisco UBE

When a dial peer has an application configured as a session application, then only the user parameter of the request URI is used and is sent from the inbound SIP SPI to the application. The session application performs a match on an outbound dial peer based on the user parameter of the request URI sent from the inbound dial peer. In the figure below, 567 is the user portion of the request-URI that is passed from the inbound dial peer to the application and the matching outbound dial-peer found is 1000.
With the introduction of the domain-based routing feature, all parameters including the domain name of the request URI will be sent to the application and the outbound dial peer can be matched with any parameter. In Figure 1, when the domain name example.com is used to match an outbound dial peer the resulting dial peer is 2000. The `call route url` command is used for configuring domain-based routing.

# How to Configure Domain-Based Routing Support on the Cisco UBE

## Configuring Domain-Based Routing at Global Level

### SUMMARY STEPS

1. enable
2. configure terminal
3. voice service voip
4. sip
5. call-route url
6. exit

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
</tbody>
</table>
### Purpose

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example: Device&gt; enable</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Step 2 configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 3 voice service voip</td>
<td>Enters voice service configuration mode.</td>
</tr>
<tr>
<td>Example: Device(config)# voice service voip</td>
<td></td>
</tr>
<tr>
<td>Step 4 sip</td>
<td>Enters voice service SIP configuration mode.</td>
</tr>
<tr>
<td>Example: Device(conf-voi-serv)# sip</td>
<td></td>
</tr>
<tr>
<td>Step 5 call-route url</td>
<td>Routes calls based on the URL.</td>
</tr>
<tr>
<td>Example: Device(conf-serv-sip)# call-route url</td>
<td></td>
</tr>
<tr>
<td>Step 6 exit</td>
<td>Exits the current mode.</td>
</tr>
<tr>
<td>Example: Device(conf-serv-sip)# exit</td>
<td></td>
</tr>
</tbody>
</table>

### Configuring Domain-Based Routing at Dial Peer Level

**SUMMARY STEPS**

1. enable
2. configure terminal
3. dial-peer voice dial-peer tag voip
4. voice-class sip call-route url
5. exit
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** enable | Enables privileged EXEC mode.  
  • Enter your password if prompted. |
| Example: Device> enable | |
| **Step 2** configure terminal | Enters global configuration mode. |
| Example: Device# configure terminal | |
| **Step 3** dial-peer voice dial-peer tag voip | Enter dial peer voice configuration mode. |
| Example: Device(config)# dial-peer voice 2 voip | |
| **Step 4** voice-class sip call-route url | |
| Example: Device(config-dial-peer)# | |
| Example: Routes calls based on the URL | |
| **Step 5** exit | Exits the current mode. |
| Example: Device(config-dial-peer)# exit | |

## Verifying and Troubleshooting Domain-Based Routing Support on the Cisco UBE

### SUMMARY STEPS

1. enable  
2. debug ccsip all  
3. debug voip dialpeer inout

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td></td>
</tr>
</tbody>
</table>

Cisco Unified Border Element Protocol-Independent Features and Setup Configuration Guide, Cisco IOS Release 15M&T
Enables privileged EXEC mode.

**Example:**
Device> enable

**Step 2 debug ccsip all**
Enables all SIP-related debugging.

**Example:**
Device# debug ccsip all

```
Received:
Via: SIP/2.0/UDP [2208:1:1:1:1:1:1:1115]:5060;branch=z9hG4bK83AE3
Date: Tue, 01 Mar 2011 08:49:48 GMT
Supported: 100rel,timer,resource-priority,replaces
Require: sdp-anat
Min-SE: 1800
Cisco-Guid: 2948477781-1125585376-2396638033-3925258737
User-Agent: Cisco-SIPGateway/IOS-15.1(3.14.2)PIA16
Allow: INVITE, OPTIONS, BYE, CANCEL, ACK, PRACK, UPDATE, REFER, SUBSCRIBE, NOTIFY, INFO, REGISTER
CSeq: 101 INVITE
Max-Forwards: 70
Timestamp: 1298969388
Expires: 180
Allow-Events: telephone-event
Content-Type: application/sdp
Content-Disposition: session;handling=required
Content-Length: 495
v=0
s=SIP Call
c=IN IP6 2208:1:1:1:1:1:1:1115
t=0 0
a-group:ANAT 1 2
m=audio 17836 RTP/AVP 0 101 19
c=IN IP6 2208:1:1:1:1:1:1:1115
a=rtmap:0 PCMU/8000
a=rtmap:101 telephone-event/8000
a=rtmap:101 telephone-event/8000
a=rtmap:101 telephone-event/8000
a=rtmap:101 telephone-event/8000
a=rtmap:101 telephone-event/8000
a=rtmap:101 telephone-event/8000
a=rtmap:101 telephone-event/8000
a=rtmap:101 telephone-event/8000
a=rtmap:101 telephone-event/8000
a=rtmap:101 telephone-event/8000
```

```
Received:
Via: SIP/2.0/UDP [2208:1:1:1:1:1:1:1116]:5060;branch=z9hG4bK38ACE
Date: Thu, 10 Feb 2011 12:35:08 GMT
Call-ID: 5DEDB77E-ADC1208-808BE770-8FCACF34@2208:1:1:1:1:1:1:1117
Supported: 100rel,timer,resource-priority,replaces,sdp-anat
Min-SE: 1800
Cisco-Guid: 1432849350-0876076256-2424621905-3925258737
```
Step 3  
**debug voip dialpeer inout**

The **debug csip all** and **debug voip dialpeer inout** commands can be entered in any order and any of the commands can be used for debugging depending on the requirement.

**Example:**

Displays information about the voice dial peers

Device# debug voip dialpeer inout

voip dialpeer inout debugging is on

The following event shows the calling and called numbers:

**Example:**

*May 1 19:32:11.731: //1-6372E2598012/DPM/dpAssociateIncomingPeerCore:
Calling Number=4085550111, Called Number=3600, Voice-Interface=0x0,
Timeout=TRUE, Peer Encap Type=ENCAP_VOIP, Peer Search Type=PEER_TYPE_VOICE,
Peer Info Type=DIALPEER_INFO_SPEECH*

The following event shows the incoming dial peer:

**Example:**

*May 1 19:32:11.731: //1-6372E2598012/DPM/dpAssociateIncomingPeerCore:
Result=Success(0) after DP_MATCH_INCOMING_DNIS; Incoming Dial-peer=100
*May 1 19:32:11.731: //1-6372E2598012/DPM/dpAssociateIncomingPeerCore:
Calling Number=4085550111, Called Number=3600, Voice-Interface=0x0,
Timeout=TRUE, Peer Encap Type=ENCAP_VOIP, Peer Search Type=PEER_TYPE_VOICE,
Peer Info Type=DIALPEER_INFO_SPEECH*

*May 1 19:32:11.731: //1-6372E2598012/DPM/dpMatchPeersCore:
Result=Success(0) after DP_MATCH_INCOMING_DNIS; Incoming Dial-peer=100
*May 1 19:32:11.735: //1-6372E2598012/DPM/dpMatchPeersCore:
Calling Number=, Called Number=3600, Peer Info Type=DIALPEER_INFO_SPEECH*

*May 1 19:32:11.735: //1-6372E2598012/DPM/dpMatchPeersCore:
Match Rule=DP_MATCH_DEST; Called Number=3600
*May 1 19:32:11.735: //1-6372E2598012/DPM/dpMatchPeersCore:
Result=Success(0) after DP_MATCH_DEST
The following event shows the matched dial peers in the order of priority:

**Example:**

List of Matched Outgoing Dial-peer(s):
1: Dial-peer Tag-3600
2: Dial-peer Tag-36

---

**Configuration Examples for Domain-Based Routing Support on the Cisco UBE**

**Example Configuring Domain-Based Routing Support on the Cisco UBE**

The following example shows how to enable domain-based routing support on the Cisco UBE:

```
Device> enable
Device# configure terminal
Device(config)# voice service voip
Device(conf-voi-serv)# sip
Device(conf-serv-sip)# call-route url
Device(conf-serv-sip)# exit
Device(config)# dial-peer voice 2 voip
Device(config-dial-peer)# voice-class sip call-route url
Device(config-dial-peer)# exit
```
CHAPTER 34

URI-Based Dialing Enhancements

The URI-Based Dialing Enhancements feature describes the enhancements made to Uniform Resource Identifier (URI)-based dialing on Cisco Unified Border Element (Cisco UBE) for Session Initiation Protocol (SIP) calls. The URI-Based Dialing Enhancements feature includes support for call routing on Cisco UBE when the user part of the incoming Request-URI is non-E164 (for example, INVITE sip:user@abc.com).

- Finding Feature Information, page 309
- Information About URI-Based Dialing Enhancements, page 309
- How to Configure URI-Based Dialing Enhancements, page 313
- Configuration Examples for URI-Based Dialing Enhancements, page 321
- Additional References for URI-Based Dialing Enhancements, page 323
- Feature Information for URI-Based Dialing Enhancements, page 323

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About URI-Based Dialing Enhancements

Cisco Unified Communications Manager (CUCM) supports dialing using directory Uniform Resource Identifiers (URIs) for call addressing. Directory URIs follow the username@host format where the host portion is an IPv4 address or a fully qualified domain name. A directory URI is a string of characters that can be used to identify a directory number. If that directory number is assigned to a phone, CUCM can route calls to that phone using the directory URI. URI dialing is available for Session Initiation Protocol (SIP) and Signaling Connection Control Part (SCCP) endpoints that support directory URIs.
The primary use of URI-based dialing is peer-to-peer calling between enterprises using complete URI addresses (that is, 'username@host'). The host part of the URI identifies the destination to which the call should be routed. In earlier Cisco Unified Border Element (Cisco UBE) URI routing, the URI was replaced in the SIP header with the destination server IP address. Then routing of calls was based on the following restrictions:

- The user part of the incoming Request-URI must be an E164 number.
- The outgoing Request-URI is always set to the session target information of the outbound dial peer.

The URI-Based Dialing Enhancements feature extends support for Cisco UBE URI-based routing of calls. With these enhancements Cisco UBE supports:

- URI-based routing when the user part of the incoming Request-URI is non-E164 (for example, INVITE sip:user@abc.com).
- URI-based routing when the user part is not present. The user part is an optional parameter in the URI (for example, INVITE sip:abc.com).
- Copying the outgoing Request-URI and To header from the inbound Request-URI and To header respectively.
- Deriving (optionally) the session target for the outbound dial peer from the host portion of the inbound URI.
- URI-based routing for 302, Refer, and Bye Also scenarios.
- Call hunting where the subsequent dial peer is selected based on URI.
- Pass through of 302, with the host part of Contact: unmodified.

### Call Flows for URI-Based Dialing Enhancements

**Case 1:** URI dialing with username being E164 or non-E164 number and Request-URI host copied from the inbound leg.

```
<table>
<thead>
<tr>
<th>UAC</th>
<th>CUBE</th>
<th>UAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>INVITE sip: <a href="mailto:user@abc.com">user@abc.com</a></td>
<td>INVITE sip: user@10.1.1.1</td>
<td>INVITE sip: user@10.1.1.1</td>
</tr>
</tbody>
</table>
| dial-peer voice 100 voip
  session protocol sipv2
  incoming uri from mydesturi
  voice-class sip call-route url
 | dial-peer voice 200 voip
  destination uri mydesturi
  session protocol sipv2
  session target ipv4:10.1.1.1
  voice-class sip requiri-passing |
```

**Case 2:** Incoming Request-URI does not contain user part. The To: header information is also copied from the peer leg when the `requiri-passing` command is enabled.
Case 3: The old behavior of setting the outbound Request-URI to session target is retained when the requiri-passing command is not enabled.

Case 4: The session target derived from the host part of the URI. The outgoing INVITE is sent to resolved IP address of the host part of the URI.

Case 5: Pass through of contact URI to request URI.
Case 6: In 302 pass-through, contact header can be passed through from one leg to another by using the `contact-passing` command.

Case 7: Pass through of refer-to URI to request URI.
How to Configure URI-Based Dialing Enhancements

Configuring Pass Through of SIP URI Headers

Perform these to configure the pass through of the host part of the Request-Uniform Resource Identifier (URI) and To Session Initiation Protocol (SIP) headers. By default, Cisco Unified Border Element (Cisco UBE) sets the host part of the URI to the value configured under the session target of the outbound dial peer. For more information, see Case 1 in the "Call Flows for URI-based Dialing Enhancements" section.
### Configuring Pass Through of Request URI and To Header URI (Global Level)

#### SUMMARY STEPS

1. enable
2. configure terminal
3. voice service voip
4. sip
5. requiri-passing
6. end

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Device&gt; enable</td>
<td></td>
</tr>
</tbody>
</table>

| **Step 2** configure terminal | Enters global configuration mode.                                        |
| Example:                    |                                                                         |
| Device# configure terminal  |                                                                         |

| **Step 3** voice service voip | Specifies VoIP encapsulation and enters voice service configuration mode. |
| Example:                     |                                                                         |
| Device(config)# voice service voip |                                                               |

| **Step 4** sip              | Enters the Session Initiation Protocol (SIP) configuration mode.         |
| Example:                    |                                                                         |
| Device(conf-voi-serv)# sip  |                                                                         |

| **Step 5** requiri-passing | Enables pass through of the host part of the Request-URI and To SIP headers. By default, Cisco UBE sets the host part of the URI to the value configured under the session target of the outbound dial peer. |
| Example:                   |                                                                         |
| Router(conf-serv-sip)# requiri-passing |                                                                 |

| **Step 6** end             | Ends the current configuration session and returns to privileged EXEC mode. |
| Example:                   |                                                                         |
| Router(conf-serv-sip)# end |                                                                         |
Configuring Pass Though of Request URI and To Header URI (Dial Peer Level)

SUMMARY STEPS

1. enable
2. configure terminal
3. voice class uri tag sip
4. host hostname-pattern
5. exit
6. dial-peer voice tag voip
7. session protocol sipv2
8. destination uri tag
9. session target ipv4:ip-address
10. voice-class sip requri-passing [system]
11. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> voice class uri tag sip</td>
<td>Creates a voice class for matching dial peers to a Session</td>
</tr>
<tr>
<td>Example:</td>
<td>Initiation Protocol (SIP) and enters voice URI class configuration mode.</td>
</tr>
<tr>
<td>Device(config)# voice class uri</td>
<td></td>
</tr>
<tr>
<td>mydesturi sip</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> host hostname-pattern</td>
<td>Matches a call based on the host field in a SIP Uniform Resource</td>
</tr>
<tr>
<td>Example:</td>
<td>Identifier (URI).</td>
</tr>
<tr>
<td>Device(config-voice-uri-class)#</td>
<td></td>
</tr>
<tr>
<td>host example.com</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> exit</td>
<td>Exits voice URI class configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device(config-voice-uri-class)#</td>
<td></td>
</tr>
<tr>
<td>exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> dial-peer voice tag</td>
<td>Defines a VoIP dial peer and enters dial peer configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device(config)# dial-peer voice</td>
<td></td>
</tr>
<tr>
<td>22 voip</td>
<td></td>
</tr>
</tbody>
</table>
### Purpose

**Step 7**  
**Command or Action:** `session protocol sipv2`  
**Example:**  
```
Device(config-dial-peer)# session protocol sipv2
```
Specifies a session protocol for calls between local and remote routers using the Internet Engineering Task Force (IETF) SIP.

**Step 8**  
**Command or Action:** `destination uri tag`  
**Example:**  
```
Device(config)# destination uri mydesturi
```
Specifies the voice class used to match a dial peer to the destination URI of an outgoing call.

**Step 9**  
**Command or Action:** `session target ipv4:ip-address`  
**Example:**  
```
Device(config-dial-peer)# session target ipv4:10.1.1.2
```
Designates a network-specific address to receive calls from a VoIP.

**Step 10**  
**Command or Action:** `voice-class sip requri-passing [system]`  
**Example:**  
```
Device(config-dial-peer)# voice-class sip requri-passing system
```
Enables the pass through of SIP URI headers.

**Step 11**  
**Command or Action:** `end`  
**Example:**  
```
Device(config-dial-peer)# end
```
Ends the current configuration session and returns to privileged EXEC mode.

---

**Configuring Pass Through of 302 Contact Header**

**Configuring Pass Through of 302 Contact Header (Global Level)**

**SUMMARY STEPS**

1. enable  
2. configure terminal  
3. voice service voip  
4. sip  
5. contact-passing  
6. end
# DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>enable</code></td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Device&gt;</code> <code>enable</code></td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td><code>configure terminal</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Device# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td><code>voice service voip</code></td>
<td>Specifies VoIP encapsulation and enters voice service configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Device(config)# voice service voip</code></td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td><code>sip</code></td>
<td>Enters voice service SIP configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Device(conf-voi-serv)# sip</code></td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td><code>contact-passing</code></td>
<td>Enables pass through of the contact header from one leg to the other leg in 302 pass through scenario.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Router(conf-serv-sip)# contact-passing</code></td>
<td></td>
</tr>
<tr>
<td>Step 6</td>
<td><code>end</code></td>
<td>Ends the current configuration session and returns to privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>Router(conf-serv-sip)# end</code></td>
<td></td>
</tr>
</tbody>
</table>
Configuring Pass Through of 302 Contact Header (Dial Peer Level)

SUMMARY STEPS

1. enable
2. configure terminal
3. voice class uri destination-tag sip
4. user-id id-tag
5. exit
6. voice service voip
7. allow-connections sip to sip
8. dial-peer voice tag voip
9. session protocol sipv2
10. destination uri destination-tag
11. voice-class sip contact-passing
12. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>Example: Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example: Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>voice class uri destination-tag sip</td>
<td>Creates a voice class for matching dial peers to a Session Initiation Protocol (SIP) and enters voice URI class configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example: Device(config)# voice class uri mydesturi sip</td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>user-id id-tag</td>
<td>Matches a call based on the User ID portion of the Uniform Resource Identifier (URI).</td>
</tr>
<tr>
<td></td>
<td>Example: Device(config-voice-uri-class)# user-id 5678</td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td>exit</td>
<td>Exits voice URI class configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example: Device(config-voice-uri-class)# exit</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
<td></td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>Specifies Voice over IP (VoIP) as the voice encapsulation type and enters voice service configuration mode.</td>
<td></td>
</tr>
<tr>
<td>voice service voip</td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Example:</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Device(config)# voice service voip</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>Allows connections between SIP endpoints in a VoIP network.</td>
<td></td>
</tr>
<tr>
<td>allow-connections sip to sip</td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Example:</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Device(conf-voi-serv)# allow-connections sip to sip</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>Defines a VoIP dial peer and enters dial peer configuration mode.</td>
<td></td>
</tr>
<tr>
<td>dial-peer voice tag voip</td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Example:</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Device(config)# dial-peer voice 200 voip</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td>Specifies a session protocol for calls between local and remote routers using the Internet Engineering Task Force (IETF) SIP.</td>
<td></td>
</tr>
<tr>
<td>session protocol sipv2</td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Example:</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Device(config-dial-peer)# session protocol sipv2</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td>Specifies the voice class used to match a dial peer to the destination URI of an outgoing call.</td>
<td></td>
</tr>
<tr>
<td>destination uri destination-tag</td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Example:</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Device(config-dial-peer)# destination uri mydesturi</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 11</strong></td>
<td>Enables pass through of the contact header from one leg to the other leg in 302 pass through scenario.</td>
<td></td>
</tr>
<tr>
<td>voice-class sip contact-passing</td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Example:</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Device(config-dial-peer)# voice-class sip contact-passing</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 12</strong></td>
<td>Ends the current configuration session and returns to privileged EXEC mode.</td>
<td></td>
</tr>
<tr>
<td>end</td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Example:</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Device(config-dial-peer)# end</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Deriving of Session Target from URI**

Perform this task to derive the session target from the host part of the Uniform Resource Identifier (URI). The outgoing INVITE is sent to the resolved IP address of the host part of the URI. For more information, see Case 4 in the "Call Flows for URI-Based Dialing Enhancements" section.
SUMMARY STEPS

1. enable
2. configure terminal
3. voice class uri destination-tag sip
4. host hostname-pattern
5. exit
6. dial-peer voice tag voip
7. session protocol sipv2
8. destination uri destination-tag
9. session target sip-uri
10. exit
11. voice class uri source-tag sip
12. host hostname-pattern
13. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>2</td>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>3</td>
<td>voice class uri destination-tag sip</td>
<td>Creates or modifies a voice class for matching dial peers to a Session Initiation Protocol (SIP) or telephone (TEL) Uniform Resource Identifier (URI) and enters voice URI class configuration mode.</td>
</tr>
<tr>
<td></td>
<td>host hostname-pattern</td>
<td>Matches a call based on the host field in a SIP URI.</td>
</tr>
<tr>
<td>4</td>
<td>dial-peer voice tag voip</td>
<td>Defines a VoIP dial peer and enters dial peer configuration mode.</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td><strong>Step 7</strong> session protocol sipv2</td>
<td>Specifies a session protocol for calls between local and remote routers using the Internet Engineering Task Force (IETF) SIP.</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config-dial-peer)# session protocol sipv2</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong> destination uri destination-tag</td>
<td>Specifies the voice class used to match a dial peer to the destination URI of an outgoing call.</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config-dial-peer)# destination uri mydesturi</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 9</strong> session target sip-uri</td>
<td>Derives session target from incoming URI.</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config-dial-peer)# session target sip-uri</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 10</strong> exit</td>
<td>Exits dial peer voice configuration mode.</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config-dial-peer)# exit</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 11</strong> voice class uri source-tag sip</td>
<td>Creates or modifies a voice class for matching dial peers to a SIP or TEL URI and enters voice URI class configuration mode.</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config)# voice class uri mysourceuri sip</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 12</strong> host hostname-pattern</td>
<td>Matches a call based on the host field in a SIP URI.</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config-voice-uri-class)# host abc.com</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step 13</strong> end</td>
<td>Ends the current configuration session and returns to privileged EXEC mode.</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config-voice-uri-class)# end</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Configuration Examples for URI-Based Dialing Enhancements

**Example: Configuring Pass Though of Request URI and To Header URI**

**Example: Configuring Pass Though of Request URI and To Header URI (Global Level)**

```
Device> enable
Device# configure terminal
Device(config)# voice service voip
```
Example: Configuring Pass Through of Request URI and To Header URI (Dial Peer Level)

```
Device(config)# voice class uri mydesturi sip
Device(config-voice-uri-class)# host xyz.com
Device(config-voice-uri-class)# exit

Device(config)# dial-peer voice 13 voip
Device(config-dial-peer)# session protocol sipv2
Device(config-dial-peer)# destination uri mydesturi
Device(config-dial-peer)# session target ipv4:10.1.1.1
Device(config-dial-peer)# voice-class sip requri-passing system
Device(config-dial-peer)# end
```

Example: Configuring Pass Through of 302 Contact Header (Global Level)

```
Device> enable
Device# configure terminal
Device(config)# voice service voip
Device(config-voi-serv)# sip
Device(config-serv-sip)# contact-passing
Device(config-serv-sip)# end
```

Example: Configuring Pass Through of 302 Contact Header (Dial Peer Level)

```
Device(config)# voice class uri mydesturi sip
Device(config-voice-uri-class)# user-id 5678
Device(config-voice-uri-class)# exit

Device(config)# dial-peer voice 200 voip
Device(config-dial-peer)# session protocol sipv2
Device(config-dial-peer)# destination uri mydesturi
Device(config-dial-peer)# voice-class sip contact-passing
Device(config-dial-peer)# end
```

Example: Deriving Session Target from URI

```
Device> enable
Device# configure terminal
Device(config)# voice class uri mydesturi sip
Device(config-voice-uri-class)# host destination.com
Device(config-voice-uri-class)# exit

Device(config)# dial-peer voice 25 voip
Device(config-dial-peer)# session protocol sipv2
Device(config-dial-peer)# destination uri mydesturi
```
Additional References for URI-Based Dialing Enhancements

### Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voice commands</td>
<td>Cisco IOS Voice Command Reference</td>
</tr>
<tr>
<td>Cisco IOS commands</td>
<td>Cisco IOS Master Command List, All Releases</td>
</tr>
<tr>
<td>SIP configuration tasks</td>
<td>SIP Configuration Guide, Cisco IOS Release 15M&amp;T</td>
</tr>
</tbody>
</table>

### Technical Assistance

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

Feature Information for URI-Based Dialing Enhancements

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 . An account on Cisco.com is not required.
Table 38: Feature Information for URI-Based Dialing Enhancements

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>URI-Based Dialing Enhancements</td>
<td>15.4(1)T</td>
<td>The URI-Based Dialing Enhancements feature includes support for call routing on Cisco UBE when the user-part of the incoming Request-URI is non-E164 (for example, INVITE sip:<a href="mailto:user@abc.com">user@abc.com</a>). The following commands were introduced or modified: contact-passing, requiri-passing, session target sip-uri and voice-class sip requiri-passing</td>
</tr>
</tbody>
</table>
Fax Detection for SIP Call and Transfer

Fax detection is the capability to detect automatically whether an incoming call is voice or fax. For calls coming from an IP trunk to the CUBE, the Fax Detection for SIP Call and Transfer feature is used to detect CNG tones (calling tones) so that the fax server can handle the actual fax transmission or redirect the fax call to a configured fax number. Once the tone is detected, the same will be reported to the session application on the incoming TDM call leg, and based on the configuration, the T.38 fax relay session is setup locally.

- Finding Feature Information, page 325
- Restrictions for Fax Detection for SIP Call and Transfer, page 325
- Information About Fax Detection for SIP Call and Transfer, page 326
- How to Configure Fax Detection for SIP Calls, page 328
- Configuration Examples for Fax Detection for SIP Calls, page 330
- Feature Information for Fax Detection for SIP Call and Transfer, page 331

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 www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for Fax Detection for SIP Call and Transfer

- For FAX detection to work, the \texttt{cng-fax-detect} command under DSP farm and the \texttt{detect-fax} command must be configured in the inbound dial-peer.
- Only the g711ulaw codec can be used for detecting fax CNG tone.
- The \texttt{cng-fax-detect} command can be configured up to maximum length of 256 characters.
The phone number can be of a maximum length of 32 characters.

SCCP-based transcoding is not supported; only LTI-based transcoder is supported.

Information About Fax Detection for SIP Call and Transfer

When a call comes in from an IP trunk to the CUBE, it loops the call to a locally present Voice XML (VXML) gateway, which establishes an auto-attended call. The CUBE monitors the incoming audio stream. The incoming call could be through a fax machine's handset and then switching to transmit a fax. When the CUBE detects a CNG tone, it can be handled in two modes based on the configuration:

- Trigger a SIP-REFER message to a remote fax server that handles the actual fax transmission.
- Redirect the fax call to a configured fax number(s) locally.

Mode 1—Local Redirect

In the local redirect mode, the call is redirected to local fax numbers and the redirect call setup will be initiated locally.

Local redirect can be configured with multiple fax numbers. The CUBE will try to set up call to the first configured fax number till the last fax number, until the call is successfully established. After a call setup is successful, the remaining fax numbers are ignored. There is no limit to the number of fax numbers that can be configured for local redirect. The maximum length of a command can be 256 characters.

Figure 17: Local Redirect Call Flow
For each call, a digital signal processor (DSP) channel or resource is allocated to detect CNG tone. In the call flow, as the first fax machine returned an error, the CUBE tries to establish the call with the second fax machine.

Note
CUBE will send out a normal VOICE SDP INVITE to the local FAX machine after the CNG tone is detected. It does not send out a FAX negotiation SDP.

**Mode 2—Refer Redirect**

In this mode, redirect through SIP-REFER message is configured for remote fax numbers.

Refer redirect can be configured with only one remote FAX number. A SIP REFER message is sent back to the incoming dial-peer to redirect the call (similar to blind transfer). The refer redirect can be configured for local fax numbers also.

*Figure 18: Refer Redirect Call Flow*

For each call, a DSP channel or resource is allocated to detect the CNG tone. Refer will be sent with the remote FAX number in the Refer-to header.
How to Configure Fax Detection for SIP Calls

Enabling CNG Fax Detection

SUMMARY STEPS

1. enable
2. configure terminal
3. dspfarm profile tag transcode universal
4. cng-fax-detect
5. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> dspfarm profile tag transcode universal</td>
<td>Enters DSP farm profile configuration mode and enables the profile for transcoding.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device(config)# dspfarm profile 5 transcode universal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> cng-fax-detect</td>
<td>Enables CNG tone detection.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device(config-dspfarm-profile)# cng-fax-detect</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> end</td>
<td>Returns to privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device(config-dspfarm-profile)# end</td>
<td></td>
</tr>
</tbody>
</table>
Verifying Fax Detection for SIP Calls

SUMMARY STEPS

1. **enable**
2. **show call active voice compact**
3. **show dspfarm dsp active**
4. **show call active voice compact**

DETAILED STEPS

**Step 1**  
**enable**

**Example:**

Device> **enable**

Enables privileged EXEC mode.

**Step 2**  
**show call active voice compact**

**Example:**

This is a sample output of call setup when the call is connected to the VXML gateway after being looped:

Device# **show call active voice compact**

```
<callID> A/O FAX T<sec> Codec type Peer Address IP R<ip>:<udp>
Total call-legs: 3
  9  ANS  T4  g711ulaw  VOIP   P808808  9.42.25.145:17940
  10 ORG  T4  g711ulaw  VOIP   P309903  9.42.25.149:16396
  11 ANS  T4  g711ulaw  VOIP   P808808  9.42.25.149:16394
```

**Step 3**  
**show dspfarm dsp active**

**Example:**

This is a sample output of the DSP channel reserved to detect CNG tone after the call is set up.

Device# **show dspfarm dsp active**

```
SLOT DSP VERSION STATUS CHNL USE TYPE RSC_ID BRIDGE_ID PKTS_TXED PKTS_RXED
  0  2  36.1.0 UP  1 USED xcode 1  9  228  119
  0  2  36.1.0 UP  1 USED xcode 1 10  113  251
Total number of DSPFARM DSP channel(s) 1
```

**Step 4**  
**show call active voice compact**

**Example:**

This is a sample output of FAX call setup with local redirect in the CUBE:

Device# **show call active voice compact**

```
<callID> A/O FAX T<sec> Codec type Peer Address IP R<ip>:<udp>
Total call-legs: 2
```

Cisco Unified Border Element Protocol-Independent Features and Setup Configuration Guide, Cisco IOS Release 15M&T
Troubleshooting Fax Detection for SIP Calls

You can enable the logs of the following `debug` or `show` commands, which are helpful in debugging fax detection for SIP calls:

- `debug ccsip verbose`
- `debug voip capi all`
- `debug voip dsmp all`
- `debug voip hpi all`
- `debug media resource provisioning all`
- `show call active voice compact`
- `show dspfarm dsp active`
- `show voip rtp connections`

Configuration Examples for Fax Detection for SIP Calls

Example: Configuring Local Redirect

In this example, three dial-peers are used. Each incoming dial-peer is associated with one service. If customers want to run their own VXML script, then can run the script in the initial incoming dial-peer (dial-peer 410 in the example below) The call is looped using translation profile to get one more incoming dial-peer in which VXML script is run for default IVR session (dial-peer 412 in the example below).

```
voice translation-rule 1 //Translation Rule//
rule 1 /903309/ /309903/

voice translation-profile vxml
translate called 1
dial-peer voice 410 voip
    description "Incoming dial-peer to GW"
    translation-profile incoming vxml
    session protocol sipv2
    incoming called-number 903309
    codec g711ulaw
    detect-fax mode local 12101 12102 12103 12104 //Local Redirect command//

dial-peer voice 411 voip
    description "Outgoing dial-peer to VXML GW"
    destination-pattern 309903
    session protocol sipv2
    session target ipv4:9.42.25.149 //CUBE IP for looping the call//
    codec g711ulaw
```
Example: Configuring Refer Redirect

In Refer mode, only one fax number can be configured.

```
dial-peer voice 410 voip
  description "Incoming dial-peer to GW"
  translation-profile incoming vxml
  translate called 1
  session protocol sipv2
  incoming called-number 903309
  codec g711ulaw

dial-peer voice 411 voip
  description "Outgoing dial-peer to VXML GW"
  destination-pattern 309903
  session protocol sipv2
  session target ipv4:9.42.25.149 //CUBE IP for looping the call
  codec g711ulaw
```

Feature Information for Fax Detection for SIP Call and Transfer

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 . An account on Cisco.com is not required.
Table 39: Feature Information for Fax Detection for SIP Call and Transfer

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fax Detection for SIP Call and Transfer</td>
<td>15.4(2)T</td>
<td>Fax detection is the capability to detect automatically whether an incoming call is voice or fax. For calls coming from an IP trunk to a the CUBE, the Fax Detection for SIP Call and Transfer feature is used to detect CNG tones (calling tones) so that the fax server can handle the actual fax transmission or redirect the fax call to a configured fax number. The following commands were introduced: <code>cng-fax-detect</code> and <code>detect-fax mode</code>.</td>
</tr>
</tbody>
</table>