SIPREC (SIP Recording)

The SIPREC (SIP Recording) feature supports media recording for Real-time Transport Protocol (RTP) streams in compliance with section 3.1.1. of RFC 7245, with CUBE acting as the Session Recording Client. SIP is used as a protocol between CUBE and the recording server. Recording of a media session is done by sending a copy of a media stream to the recording server. Metadata is the information that is passed by the recording client to the recording server in a SIP session. The recording metadata describes the communication session and its media streams, and also identifies the participants of the call. CUBE acts as the recording client and any third party recorder acts as the recording server.

Feature Information for SIPREC-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 Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Feature Name	Releases	Feature Information
SIPREC (SIP Recording)	Cisco IOS 15.6(1)T Cisco IOS XE 3.17S	The SIPREC Recording feature supports recording of audio and video calls. Only audio and video media lines are forked. The following commands were modified: **recorder parameter** and **recorder profile**.

Prerequisites for SIPREC Recording

Make sure that:

- Recorders must be reachable from CUBE
- SIPREC should be configured; else, CUBE will fall back to the existing Network-Based Recording implementation. For more information, see *Network-Based Recording* section.
- CUBE supports the SIP Recording Metadata model format requirements specified in draft-ietf-siprec-metadata-17. Recorders must support metadata format of ver17 at a minimum
- CUBE should be in compliance with the Session Recording Protocols defined in draft-ietf-siprec-protocol-16. CUBE supports only the “siprec Option” Tag and the “src feature” tag among the various other extensions defined in the protocols draft; CUBE does not support the SDP extensions.

Restrictions for SIPREC Recording

SIPREC-based recording is not supported for the following calls:

- Any media service parameter change via Re-INVITE or UPDATE from recording server is not supported. For example, hold-resume or any codec changes
- IPv6-to-IPv6 call recording
- IPv6-to-IPv4 call recording if the recording server is configured on the IPv6 call leg
- 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
- Secure Real-time Transport Protocol (SRTP) passthrough calls
- SRTP-RTP calls with forking for SRTP leg (forking is supported for the RTP leg)
- Multicast music on hold (MOH)
- Mid-call renegotiation and supplementary services like Hold/Resume, control pause, and so on are not supported on the recorder call leg
- Recording is not supported if CUBE is running a TCL IVR application with the exception of survivability.tcl, which is supported with SIPREC based recording
- Media mixing on forked streams is not supported
- Digital Signal Processing (DSP) resources are not supported on forked legs
- Server Groups in outbound dial-peers towards recorders is not supported.

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

Information About SIPREC Recording Using CUBE

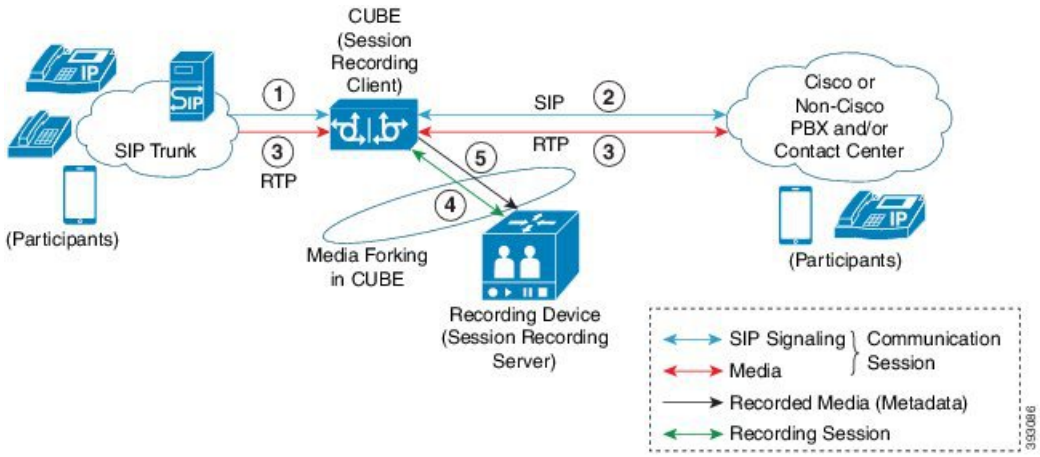
Deployment

You need to have:

- Participants — SIP UAs involved in the Communication Session. The UA can be any SIP element.
- Communication Session (CS) — Session established between the endpoints.
- Session Recording Client (SRC) — CUBE acts as the session recording client that triggers the recording session.
- Session Recording Server (SRS) — A SIP User Agent (UA) which is a specialized media server and that acts as a sink for the recorded media and metadata.
- Recording Session (RS) — SIP dialog established between CUBE (recording client) and the recording server.
- Recording Metadata — Information on the CS and the associated media stream data sent from CUBE to RS.

The following figure illustrates a third party recorder deployment with CUBE.

Figure 1: Deployment Scenario for SIPREC Recording Solution



Information flow is described below:

1. Incoming call from SIP trunk

2. Outbound call to Contact Center
3. Media between endpoints flowthrough CUBE
4. CUBE sets up a new SIP session with the recording device (SRS)
5. CUBE forks RTP media to SRS

In the preceding illustration, the Real Time Protocol (RTP) carries voice data and media streams between the user agents and CUBE. The RTP unidirectional stream represent the communication session forked from CUBE to the recording server to indicate forked media. The Session Initiation protocol (SIP) carries call signaling information along with the metadata information. Media streams from CUBE to recording server are unidirectional because only CUBE sends recorded data to recording server; the recording server does not send any media to CUBE.

Metadata has the following functions:

- Carry the communication session data (audio and video calls) that describes the call to the recording server.
- Identifies the participants list.
- Identifies the session and media association time.

If there are any changes in the call sessions, for example, hold-resume, transfer and so on, these sessions are notified to the recording server through metadata.

SIPREC High Availability Support

High availability is supported for SIPREC recording using CUBE. All metadata elements will be checkpointed in a forked call when high-availability is configured. In the event of SSO, all the forked calls and media contexts are preserved on failover.

How to Configure SIPREC-Based Recording

Configuring SIPREC-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** *profile-tag* **siprec**
9. **exit**
10. **dial-peer voice** *dp-tag* **voip**

11. **session protocol sipv2**
12. **media-class** *tag*
13. **dial-peer voice** *dial-peer-tag* **voip**
14. **destination-pattern** [+] *string* [**T**]
15. **session protocol sipv2**
16. **session target ipv4:**[*recording-server-destination-address* | *recording-server-dns*]
17. **session transport tcp**
18. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**media profile recorder** *profile-tag* **Example:** `Device(config)# media profile recorder 100`	Configures the media profile recorder and enters media profile configuration mode.
Step 4	(Optional) **media-type audio** **Example:** `Device(cfg-mediaprofile)# media-type audio`	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.
Step 5	**media-recording** *dial-peer-tag* [*dial-peer-tag2...dial-peer-tag5*] **Example:** `Device(cfg-mediaprofile)# media-recording 8000 8001 8002`	Configures the dial-peers that need to be configured **Note** You can specify a maximum of five dial-peer tags.
Step 6	**exit** **Example:** `Device(cfg-mediaprofile)# exit`	Exits media profile configuration mode.
Step 7	**media class** *tag* **Example:** `Device(config)# media class 100`	Configures a media class and enters media class configuration mode.

	Command or Action	Purpose
Step 8	**recorder profile** *profile-tag* **siprec** **Example:** `Device(cfg-mediaclass)# recorder profile 100 siprec`	Configures the media profile SIPREC recorder.
Step 9	**exit** **Example:** `Device(cfg-mediaclass)# exit`	Exits media class configuration mode.
Step 10	**dial-peer voice** *dp-tag* **voip** **Example:** `Device(config)# dial-peer voice 1 voip`	Dial peer that needs to be forked.
Step 11	**session protocol sipv2** **Example:** `Device(config-dial-peer)# session protocol sipv2`	Configures the VoIP dial peer to use Session Initiation Protocol (SIP).
Step 12	**media-class** *tag* **Example:** `Device(config-dial-peer)# media-class 100`	Configures media class on a dial peer.
Step 13	**dial-peer voice** *dial-peer-tag* **voip** **Example:** `Device(config)# dial-peer voice 8000 voip`	Configures a recorder dial peer and enters dial peer voice configuration mode.
Step 14	**destination-pattern** [+] *string* [**T**] **Example:** `Device(config-dial-peer)# destination-pattern 595959`	Specifies either the prefix or the full E.164 telephone number (depending on your dial plan) to be used for a dial peer.
Step 15	**session protocol sipv2** **Example:** `Device(config-dial-peer)# session protocol sipv2`	Configures the VoIP dial peer to use Session Initiation Protocol (SIP).
Step 16	**session target ipv4:**[*recording-server-destination-address* | *recording-server-dns*] **Example:**	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*

	Command or Action	Purpose
	`Device(config-dial-peer)# session target ipv4:10.42.29.7`	
Step 17	**session transport tcp** **Example:** `Device(config-dial-peer)# session transport tcp`	Configures a VoIP dial peer to use Transmission Control Protocol (TCP).
Step 18	**end** **Example:** `Device(config-dial-peer)# end`	Returns to privileged EXEC mode.

Configuring SIPREC-Based Recording (without Media Profile Recorder)

SUMMARY STEPS

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

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**media class** *tag* **Example:** `Device(config)# media class 100`	Configures the media class and enters media class configuration mode.
Step 4	**recorder parametersiprec** **Example:** `Device(cfg-mediaclass)# recorder parameter siprec`	Enables SIPREC recording.
Step 5	(Optional) **media-type audio** **Example:** `Device(cfg-mediaprofile)# media-type audio`	Configures recording of audio only in a call with both audio and video. **Note** If this configuration is not done, both audio and video are recorded.
Step 6	**media-recording** *dial-peer-tag* **Example:** `Device(cfg-mediaclass-recorder)# media-recording 8000, 8001, 8002`	Configures voice-class recording parameters. **Note** You can specify a maximum of five dial-peer tags.
Step 7	**exit** **Example:** `Device(cfg-mediaclass-recorder)# exit`	Exits media class recorder parameter configuration mode.
Step 8	**exit** **Example:** `Device(cfg-mediaclass)# exit`	Exits media class configuration mode.
Step 9	**dial-peer voice** *dp-tag* **voip** **Example:** `Device(config)# dial-peer voice 1 voip`	Dial peer that needs to be forked.
Step 10	**session protocol sipv2** **Example:** `Device(config-dial-peer)# session protocol sipv2`	Configures the VoIP dial peer to use Session Initiation Protocol (SIP).

	Command or Action	Purpose
Step 11	**media-class** *tag* **Example:** `Device(config-dial-peer)# media-class 100`	Configures media class on a dial peer.
Step 12	**dial-peer voice** *dial-peer-tag* **voip** **Example:** `Device(config)# dial-peer voice 8000 voip`	Configures a recorder dial peer and enters dial peer voice configuration mode.
Step 13	**destination-pattern** [+] *string* [**T**] **Example:** `Device(config-dial-peer)# destination-pattern 595959`	Specifies either the prefix or the full E.164 telephone number (depending on your dial plan) to be used for a dial peer.
Step 14	**session protocol sipv2** **Example:** `Device(config-dial-peer)# session protocol sipv2`	Configures the VoIP dial peer to use Session Initiation Protocol (SIP).
Step 15	**session target ipv4:**[*recording-server-destination-address* | *recording-server-dns*] **Example:** `Device(config-dial-peer)# session target ipv4:10.42.29.7`	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*
Step 16	**session transport tcp** **Example:** `Device(config-dial-peer)# session transport tcp`	Configures a VoIP dial peer to use Transmission Control Protocol (TCP).
Step 17	**end** **Example:** `Device(config-dial-peer)# end`	Returns to privileged EXEC mode.

Configuration Examples for SIPREC-based Recording

Example: Configuring SIPREC-based Recording with Media Profile Recorder

```
Router> enable
Router# configure terminal
```

```
Router(config)# media class 101
Router(cfg-mediaclass)# recorder profile 201 siprec
```

Example: Configuring SIPREC-based Recording without Media Profile Recorder

```
Router> enable
Router# configure terminal
Router(config)# media class 101
Router(cfg-mediaclass)# recorder parameter siprec
Router(cfg-mediaclass-recorder)# media-recording 403
```

Validate SIPREC Functionality

Use the command **show voip rtp connections** to verify that media forking configuration is correct:

```
CUBE#show voip rtp connections
VoIP RTP active connections :
No. CallId dstCallId  LocalRTP   RmtRTP     LocalIP        RemoteIP
1    36     37         18358      19362    209.165.201.5  209.165.201.10
2    37     36         17294      17690    10.0.0.5       10.0.0.20
3    39     38         19812      42196    172.16.0.10    10.0.0.10
4    40     38         24230      60234    172.16.0.10    10.0.0.10
Found 4 active RTP connections
```

In this example, the call between the 2 phones has resulted into 2 RTP streams (1 and 2). The 2 RTP streams (3 and 4) are the recorded streams that are sent to the Recording Server (10.0.0.10 in this example). The call Recording Server receives a duplicated RTP stream that represents the recorded call. Use the command show voip recmsp session to verify:

```
CUBE#sh voip recmsp session
RECMSP active sessions:
MSP Call-ID    AnchorLeg Call-ID  ForkedLeg Call-ID
  143                 141            145
Found 1 active sessions
```

To get more details of the streams run the command show voip recmsp session detail call-id <the value specified in the above op>:

```
CUBE#show voip recmsp session detail call-id <the value specified in the above o/p>
CUBE#show voip recmsp session detail call-id 143
RECMSP active sessions:
Detailed Information
========================
Recording MSP Leg Details:
Call ID: 143
GUID : 7C5946D38ECD
AnchorLeg Details:
Call ID: 141
Forking Stream type: voice-nearend
Participant: 2001
Non-anchor Leg Details:
Call ID: 140
Forking Stream type: voice-farend
Participant: 1001
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
```

Where:

- Stream State: This state shows the state of the call – can be either in ACTIVE or HOLD state.
- Anchor Leg Call-id: This ID is the call-id of the anchor leg (Dial-peer where forking is enabled) which in also internal to the system. The output in brief describes the participant number and stream type as voice near-end, which is called party side.
- Non-Anchor Call-id: This ID is the call-id of nonanchor leg (Dial-peer where forking is not enabled).
- Forked Call-id: This forking leg call-id shows near-end and far-end stream call-id details with state of the Stream.

If you want to know the remote IPs and ports for the near-end and far-end legs, use the **show voip rtp forking** command:

```
CUBE#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 10.0.0.10, 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 10.0.0.10, 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
```

Remote IP/ Port is the recording server ip and port address. Codec indicates which codec is negotiated to record the call leg. Packets that are sent indicate the number of packets that are sent to Recording Server from each stream.

Troubleshoot

The following is a sample SIPREC configuration on IOS/IOS-XE voice routers.

```
media class 777
recorder parameter siprec
media-recording 777
!
dial-peer voice 11 voip
description CUCM
session protocol sipv2
session target ipv4:10.0.0.15
destination e164-pattern-map 164
media-class 777
codec g711ulaw
!
dial-peer voice 777 voip
destination-pattern AAAA
session protocol sipv2
session target ipv4:10.0.0.10
codec g711ulaw
```

Working Scenario

After the call is connected, the inbound/outbound CCS SIP info legs helps to understand that a recording call has been initiated. In the following example, the outbound call leg 4536 posts a media forking start indication to its peer inbound leg 4535. This inbound leg ignores this event because it is not the anchor leg (in this example, media-class command is configured on the outgoing dial peer (Peer ID 4536)).

```
017895: May 13 15:32:45.273:
//4536/2FD863BAA01F/SIP/Info/info/32768/ccsip_trigger_media_forking: MF: EO leg. set the
pending
flag. wait for peer leg to indicate start
017896: May 13 15:32:45.273:
//4536/2FD863BAA01F/SIP/Info/info/32768/ccsip_trigger_media_forking: MF: posting
CC_EV_H245_MEDIA_FORKING_START_IND.
017901: May 13 15:32:45.273: //4535/2FD863BAA01F/SIP/Info/notify/32768/ccsip_event_handler:
CC_EV_H245_MEDIA_FORKING_START_IND: peer ID 4536, event = 217 type = 1
017902: May 13 15:32:45.273: //4535/2FD863BAA01F/SIP/Info/verbose/32768/ccsip_event_handler:
Ignoring the event on non-anchor leg
```

Similarly, the outbound call leg 4536 posts a media forking start indication to the inbound call leg 5435.

```
018221: May 13 15:32:45.290: //4536/2FD863BAA01F/SIP/Info/notify/32768/ccsip_event_handler:
CC_EV_H245_MEDIA_FORKING_START_IND: peer ID 4535, event = 217 type = 1
```

Outbound leg processes the event and triggers the recording session.

```
018222: May 13 15:32:45.290: //4536/2FD863BAA01F/SIP/Info/verbose/32768/ccsip_event_handler:
Peer leg has indicated start. Trigger Media Forking.
```

```
018229: May 13 15:32:45.290: //-1/xxxxxxxxxxxx/Event/recmsp_api_create_session: Event:
E_REC_CREATE_SESSION anchor call ID:4536, msp call ID:4537
018230: May 13 15:32:45.290: //-1/xxxxxxxxxxxx/Inout/recmsp_api_create_session: Exit with
Success
```

Recording dial-peer lookup.

```
018320: May 13 15:32:45.293: //4537/2FD863BAA01F/RECMSP/Inout/recmsp_get_dp_tag_list: REC
DP: =
777
018390: May 13 15:32:45.296: //-
1/xxxxxxxxxxxx/SIP/Info/verbose/5120/sipSPIGetOutboundHostAndDestHostPrivate: CCSIP:
target_host
: 10.0.0.10 target_port : 5060
```

Create XML metadata.

```
018513: May 13 15:32:45.301:
//4538/2FD863BAA01F/SIP/Info/info/32768/ccsip_ipip_mf_create_xml_metadata: MF: XML metadata
 Len:
[1763]
<?xml version="1.0" encoding="UTF-8"?>
<recording xmlns="urn:ietf:params:xml:ns:recording:1">
<datamode>complete</datamode>
<session session_id="MIgZ2nTLEemWFaQi1vyb4Q==">
<sipSessionID>a0b9b2a1e4db51f082e777c0df9015e5;remote=6bea155500105000a0002c31246a214b</sipSessi
onID>
<start-time>2019-05-13T15:32:45.293Z</start-time>
</session>
<participant participant_id="MIhBMXTLEemWFqQi1vyb4Q==">
<nameID aor="sip:1234@10.0.0.15">
</nameID>
</participant>
<participantses**MSG 00003 TRUNCATED**
**MSG 00003 CONTINUATION #01**sionassoc participant_id="MIhBMXTLEemWFqQi1vyb4Q=="
session_id="MIgZ2nTLEemWFaQi1vyb4Q==">
<associate-time>2019-05-13T15:32:45.293Z</associate-time>
```

```
</participantsessionassoc>
<stream stream_id="MIlSKnTLEemWG6Qi1vyb4Q==" session_id="MIgZ2nTLEemWFaQi1vyb4Q==">
<label>1</label>
</stream>
<participant participant_id="MIhBMXTLEemWF6Qi1vyb4Q==">
<nameID aor="sip:911@209.165.201.1">
<name xml:lang="en">Emergency</name>
</nameID>
</participant>
<participantsessionassoc participant_id="MIhBMXTLEemWF6Qi1vyb4Q=="
session_id="MIgZ2nTLEemWFaQi1vyb4Q==">
<asso**MSG 00003 TRUNCATED**
**MSG 00003 CONTINUATION #02**ciate-time>2019-05-13T15:32:45.293Z</associate-time>
</participantsessionassoc>
<stream stream_id="MIlSKnTLEemWHKQi1vyb4Q==" session_id="MIgZ2nTLEemWFaQi1vyb4Q==">
<label>2</label>
</stream>
<participantstreamassoc participant_id="MIhBMXTLEemWFqQi1vyb4Q==">
<send>MIlSKnTLEemWG6Qi1vyb4Q==</send>
<recv>MIlSKnTLEemWHKQi1vyb4Q==</recv>
</participantstreamassoc>
<participantstreamassoc participant_id="MIhBMXTLEemWF6Qi1vyb4Q==">
<send>MIlSKnTLEemWHKQi1vyb4Q==</send>
<recv>MIlSKnTLEemWG6Qi1vyb4Q==</recv>
</participantstreamassoc>
</recording>
```

INVITE is sent to recorder with metadata in XML format where:

- The **nameID** attribute represents the name and SIP/SIPS/tel URI (also called the address of record) of each participant.
- The **participant_id** attribute indicates the unique ID assigned to each participant in the recording session.
- The **stream_id** attribute indicates the unique ID assigned to each media stream in the recording session.
- The **session_id** attribute is used to reference the communication session to which a given media stream belongs.
- The label metadata attribute provides the value of **a=label** attribute assigned to this media stream in the SDP of the SIP request and responses of the recording session. It plays a key role in associating a media stream with its metadata information.

```
018628: May 13 15:32:45.306: //4538/2FD863BAA01F/SIP/Msg/ccsipDisplayMsg:
Sent:
INVITE sip:AAAA@10.0.0.10:5060 SIP/2.0
Via: SIP/2.0/UDP y.y.y.y:5060;branch=z9hG4bK11BD2CA
From: <sip:y.y.y.y>;tag=F75AD7F-2065
To: <sip:AAAA@10.0.0.10>
Date: Mon, 13 May 2019 15:32:45 GMT
Call-ID: 3089C795-74CB11E9-961DA422-D6FC9BE1@y.y.y.y
Supported: 100rel,timer,resource-priority,replaces,sdp-anat
Require: siprec
Min-SE: 1800
Cisco-Guid: 0802710458-1959465449-2686421522-1015028268
User-Agent: Cisco-SIPGateway/IOS-16.10.2
Allow: INVITE, OPTIONS, BYE, CANCEL, ACK, PRACK, UPDATE, REFER, SUBSCRIBE, NOTIFY, INFO,
REGISTER
CSeq: 101 INVITE
Max-Forwards: 70
Timestamp: 1557761565
Contact: <sip:y.y.y.y:5060>;+sip.src
Expires: 180
```

```
Allow-Events: telephone-event
Session-ID: a62dd6d8be0059c38d142bae9b46880b;remote=00000000000000000000000000000000
Session-Expires: 1800
Content-Type: multipart/mixed;boundary=uniqueBoundary
Mime-Version: 1.0
Content-Length: 2470
--uniqueBoundary
  Content-Type: application/sdp
  Content-Disposition: session;handling=required
  v=0
  o=CiscoSystemsSIP-GW-UserAgent 5511 2889 IN IP4 y.y.y.y
  s=SIP Call
  c=IN IP4 y.y.y.y
  t=0 0
  m=audio 8086 RTP/AVP 0 101 19
  c=IN IP4 y.y.y.y
  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
  a=label:1
  m=audio 8088 RTP/AVP 0 101 19
  c=IN IP4 y.y.y.y
  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
  a=label:2
--uniqueBoundary
 Content-Type: application/rs-metadata+xml
 Content-Disposition: recording-session
 <?xml version="1.0" encoding="UTF-8"?>
 <recording xmlns="urn:ietf:params:xml:ns:recording:1">
   <datamode>complete</datamode>
   <session session_id="MIgZ2nTLEemWFaQi1vyb4Q==">

<sipSessionID>a0b9b2a1e4db51f082e777c0df9015e5;remote=6bea155500105000a0002c31246a214b</sipSessi

   onID>
   <start-time>2019-05-13T15:32:45.293Z</start-time> </session>
   <participant participant_id="MIhBMXTLEemWFqQi1vyb4Q==">
   <nameID aor="sip:1234@10.0.0.15">
   </nameID>
   </participant>
   <participantsessionassoc participant_id="MIhBMXTLEemWFqQi1vyb4Q=="
      session_id="MIgZ2nTLEemWFaQi1vyb4Q==">
      <associate-time>2019-05-13T15:32:45.293Z</associate-time>
   </participantsessionassoc>
   <stream stream_id="MIlSKnTLEemWG6Qi1vyb4Q==" session_id="MIgZ2nTLEemWFaQi1vyb4Q==">
   <label>1</label>
   </stream>
   <participant participant_id="MIhBMXTLEemWF6Qi1vyb4Q==">
      <nameID aor="sip:911@209.165.201.1">
      <name xml:lang="en">Emergency</name>
      </nameID>
   </participant>
   <participantsessionassoc participant_id="MIhBMXTLEemWF6Qi1vyb4Q==="
      session_id="MIgZ2nTLEemWFaQi1vyb4Q==">
      <associate-time>2019-05-13T15:32:45.293Z</associate-time>
   </participantsessionassoc>
```

```
   <stream stream_id="MIlSKnTLEemWHKQi1vyb4Q==" session_id="MIgZ2nTLEemWFaQi1vyb4Q==">
      <label>2</label>
   </stream>
   <participantstreamassoc participant_id="MIhBMXTLEemWFqQi1vyb4Q==">
     <send>MIlSKnTLEemWG6Qi1vyb4Q==</send>
     <recv>MIlSKnTLEemWHKQi1vyb4Q==</recv>
   </participantstreamassoc>
   <participantstreamassoc participant_id="MIhBMXTLEemWF6Qi1vyb4Q==">
     <send>MIlSKnTLEemWHKQi1vyb4Q==</send>
     <recv>MIlSKnTLEemWG6Qi1vyb4Q==</recv>
   </participantstreamassoc>
 </recording>
--uniqueBoundary--
```

In 200 OK recorder sends media **a=recvonly** and media forking is started.

```
018638: May 13 15:32:45.307: //4538/2FD863BAA01F/SIP/Msg/ccsipDisplayMsg:
Received:
SIP/2.0 200 OK
Via: SIP/2.0/UDP y.y.y.y:5060;branch=z9hG4bK11BD2CA
From: <sip:y.y.y.y>;tag=F75AD7F-2065
To: <sip:AAAA@10.0.0.10>;tag=7
Call-ID: 3089C795-74CB11E9-961DA422-D6FC9BE1@y.y.y.y
CSeq: 101 INVITE
Contact: <sip:10.0.0.10:5060;transport=UDP>
Content-Type: application/sdp
Content-Length: 207
v=0
o=user1 53655765 2353687637 IN IP4 10.0.0.10
s=-
c=IN IP4 10.0.0.10
t=0 0
m=audio 6000 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=recvonly
m=audio 8002 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=recvonly
018809: May 13 15:32:45.313: //4537/2FD863BAA01F/RECMSP/Event/recmsp_api_connect: Event:
E_REC_CC_CONNECTmsp call ID:4537 in recmsp_api_connect
```

Configuration Example for Metadata Variations with Different Mid-call Flows

Example: Complete SIP Recording Metadata Information Sent in INVITE or Re-INVITE

The following example provides all the elements involved in Recording Metadata XML body.

```
--uniqueBoundary
Content-Type: application/sdp
Content-Disposition: session;handling=required
v=0
o=CiscoSystemsSIP-GW-UserAgent 509 7422 IN IP4 9.42.25.149
s=SIP Call
c=IN IP4 9.42.25.149
t=0 0
m=audio 16552 RTP/AVP 8 101
```

```
c=IN IP4 9.42.25.149
a=rtpmap:8 PCMA/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=ptime:20
a=sendonly
a=label:1
m=audio 16554 RTP/AVP 8 101
c=IN IP4 9.42.25.149
a=rtpmap:8 PCMA/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=ptime:20
a=sendonly
a=label:2
m=video 16556 RTP/AVP 119
c=IN IP4 9.42.25.149
b=TIAS:1000000
a=rtpmap:119 H264/90000
a=fmtp:119 profile-level-id=42801E;packetization-mode=0
a=sendonly
a=label:3
m=video 16558 RTP/AVP 97
c=IN IP4 9.42.25.149
b=TIAS:1000000
a=rtpmap:97 H264/90000
a=fmtp:97 profile-level-id=42801E;packetization-mode=0
a=sendonly
a=label:4
--uniqueBoundary
Content-Type: application/rs-metadata+xml
Content-Disposition: recording-session

<?xml version="1.0" encoding="UTF-8"?>
<recording xmlns="urn:ietf:params:xml:ns:recording:1">
    <datamode>complete</datamode>
    <session session_id="JaPQeP1CEeSA66sYHx7YVg==">
    <sipSessionID>276ac102a3c05270a4375d99512ea1a1;remote=110b0c0f50775078b13d60be0044db11
    </sipSessionID>
    <start-time>2015-05-19T09:42:06.911Z</start-time>
    </session>
    <participant participant_id="JaPQeP1CEeSA76sYHx7YVg==">
        <nameID aor="sip:808808@9.0.0.174">
            <name xml:lang="en">808808</name>
        </nameID>
    </participant>
    <participantsessionassoc participant_id="JaPQeP1CEeSA76sYHx7YVg=="
session_id="JaPQeP1CEeSA66sYHx7YVg==">
        <associate-time>2015-05-19T09:42:06.911Z</associate-time>
    </participantsessionassoc>
    <stream stream_id="JaPQeP1CEeSA8KsYHx7YVg==" session_id="JaPQeP1CEeSA66sYHx7YVg==">
        <label>1</label>
    </stream>
    <stream stream_id="JaPQeP1CEeSA8asYHx7YVg==" session_id="JaPQeP1CEeSA66sYHx7YVg==">
        <label>3</label>
    </stream>
    <participant participant_id="JaPQeP1CEeSA8qsYHx7YVg==">
        <nameID aor="sip:909909@9.0.0.174">
            <name xml:lang="en">909909</name>
        </nameID>
    </participant>
    <participantsessionassoc participant_id="JaPQeP1CEeSA8qsYHx7YVg==="
session_id="JaPQeP1CEeSA66sYHx7YVg==">
        <associate-time>2015-05-19T09:42:06.911Z</associate-time>
```

```
    </participantsessionassoc>
    <stream stream_id="JaPQeP1CEeSA86sYHx7YVg==" session_id="JaPQeP1CEeSA66sYHx7YVg==">
        <label>2</label>
    </stream>
    <stream stream_id="JaPQeP1CEeSA9KsYHx7YVg==" session_id="JaPQeP1CEeSA66sYHx7YVg==">
        <label>4</label>
    </stream>
    <participantstreamassoc participant_id="JaPQeP1CEeSA76sYHx7YVg==">
        <send>JaPQeP1CEeSA8KsYHx7YVg==</send>
        <recv>JaPQeP1CEeSA86sYHx7YVg==</recv>
        <send>JaPQeP1CEeSA8asYHx7YVg==</send>
        <recv>JaPQeP1CEeSA9KsYHx7YVg==</recv>
    </participantstreamassoc>
    <participantstreamassoc participant_id="JaPQeP1CEeSA8qsYHx7YVg==">
        <send>JaPQeP1CEeSA86sYHx7YVg==</send>
        <recv>JaPQeP1CEeSA8KsYHx7YVg==</recv>
        <send>JaPQeP1CEeSA9KsYHx7YVg==</send>
        <recv>JaPQeP1CEeSA8asYHx7YVg==</recv>
    </participantstreamassoc>
</recording>
—uniqueBoundary—
```

Output Field	Description
urn:ietf:params:xml:ns:recording:1	Defines the namespace URI for the elements—Uniform Resource Namespace (URN).
datamode>complete</datamode	<dataMode> is a recording element that indicates whether the XML document is a complete document or a partial update. If no <dataMode> element is present then the default value is "complete".
session session_id="JaPQeP1CEeSA66sYHx7YVg=="	Session ID which remains constant for the complete call leg.
sipSessionID 276ac102a3c05270a4375d99512ea1a1; remote=110b0c0f50775078b13d60be0044db11	This attribute carries a SIP Session-ID of the original call between the participants.
<participant participant_id="JaPQeP1CEeSA76sYHx7YVg=="> <nameID aor="sip:808808@9.0.0.174">	Name and participant ID of the first participant. The first participant will always be the anchor leg of the call. Each participant has a unique 'participant_id' attribute. For example, nameID is **sip:808808**.
a=**label:1**; <stream stream_id="JaPQeP1CEeSA86sYHx7YVg==" session_id="JaPQeP1CEeSA66sYHx7YVg==">**<label>1</label>** </stream>	The <stream> element represents a Media Stream object. Stream element indicates the SDP media lines associated with the session and participants. The <label> element within the <stream> element references an SDP "a=label" attribute that identifies an m-line within the RS SDP. This m-line carries the media stream from the SRC to the SRS.

Output Field	Description
participantsessionassoc participant_id="JaPQeP1CEeSA76sYHx7YVg==" session_id="JaPQeP1CEeSA66sYHx7YVg==">>	Participant CS Association class describes the association of the first participant to a CS for a period of time. A participant can associate and dissociate from a CS several times. ParticipantCS association class has the following attributes: • Associate-time—Time when the participant is associated to CS. • Disassociate-time—Time when the participant is disassociated from a CS. Each CS object is represented by one session element. Each session element has a unique 'session_id' attribute which helps to identify unique CS sessions.
participantsessionassoc participant_id="JaPQeP1CEeSA8qsYHx7YVg==" session_id="JaPQeP1CEeSA66sYHx7YVg==">>	Participant CS Association class describes the association of the second participant to a CS for a period of time. A participant can associate and dissociate from a CS several times. The 'session_id' attribute helps to identify unique CS session of the second participant.
participantstreamassoc participant_id="JaPQeP1CEeSA76sYHx7YVg==">; participantstreamassoc participant_id="JaPQeP1CEeSA8qsYHx7YVg==">;	Participant stream association class describes the association of either participant 1 or 2 to a media stream for a period of time, as a sender or as a receiver, or both. These streams can be either audio or video or both. ParticipantStream association class has the following attributes: • Associate-time—Time when the participant starts contributing for a media stream. • Disassociate-time—Time when the participant stops receiving a media stream.

Example: Hold with Send-only / Recv-only Attribute in SDP

When a participant puts the audio call on hold with send-only attribute, the stream is sent only in one direction.

Here, in a normal recording session, both participants sent audio and video streams.

```
--uniqueBoundary
Content-Type: application/sdp
Content-Disposition: session;handling=required

v=0
o=CiscoSystemsSIP-GW-UserAgent 2973 4879 IN IP4 9.42.25.149
s=SIP Call
c=IN IP4 9.42.25.149
```

```
t=0 0
m=audio 16464 RTP/AVP 0 101
c=IN IP4 9.42.25.149
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=ptime:20
a=sendonly
a=label:1
m=audio 16466 RTP/AVP 0 101
c=IN IP4 9.42.25.149
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=ptime:20
a=sendonly
a=label:2
m=video 16468 RTP/AVP 97
c=IN IP4 9.42.25.149
b=TIAS:1000000
a=rtpmap:97 H264/90000
a=fmtp:97 profile-level-id=42801E;packetization-mode=0
a=sendonly
a=label:3
m=video 16470 RTP/AVP 97
c=IN IP4 9.42.25.149
b=TIAS:1000000
a=rtpmap:97 H264/90000
a=fmtp:97 profile-level-id=42801E;packetization-mode=0
a=sendonly
a=label:4

--uniqueBoundary
Content-Type: application/rs-metadata+xml
Content-Disposition: recording-session

<?xml version="1.0" encoding="UTF-8"?>
<recording xmlns="urn:ietf:params:xml:ns:recording:1">
…
    <stream stream_id="jIBTUf1BEeSAdKsYHx7YVg==" session_id="jH+2kf1BEeSAb6sYHx7YVg==">
        <label>1</label>
    </stream>
    <stream stream_id="jIBTUf1BEeSAdasYHx7YVg==" session_id="jH+2kf1BEeSAb6sYHx7YVg==">
        <label>3</label>
    </stream>
…
    <stream stream_id="jIBTUf1BEeSAd6sYHx7YVg==" session_id="jH+2kf1BEeSAb6sYHx7YVg==">
        <label>2</label>
    </stream>
    <stream stream_id="jIBTUf1BEeSAeKsYHx7YVg==" session_id="jH+2kf1BEeSAb6sYHx7YVg==">
        <label>4</label>
    </stream>
    <participantstreamassoc participant_id="jIBTUf1BEeSAc6sYHx7YVg==">
        <send>jIBTUf1BEeSAdKsYHx7YVg==</send>
        <recv>jIBTUf1BEeSAd6sYHx7YVg==</recv>
        <send>jIBTUf1BEeSAdasYHx7YVg==</send>
        <recv>jIBTUf1BEeSAeKsYHx7YVg==</recv>
    </participantstreamassoc>
    <participantstreamassoc participant_id="jIBTUf1BEeSAdqsYHx7YVg==">
        <send>jIBTUf1BEeSAd6sYHx7YVg==</send>
        <recv>jIBTUf1BEeSAdKsYHx7YVg==</recv>
        <send>jIBTUf1BEeSAeKsYHx7YVg==</send>
        <recv>jIBTUf1BEeSAdasYHx7YVg==</recv>
    </participantstreamassoc>
```

```
</recording>

--uniqueBoundary--
```

In this scenario, the second participant puts the call on hold using sendonly and the first participant will respond using recvonly. You can see from the **`participantStream association`** element that the second participant only sends audio and video streams and the first participant just receives the media streams.

The output after the second participant puts the call on hold with sendonly attribute:

```
--uniqueBoundary
Content-Type: application/sdp

v=0
o=CiscoSystemsSIP-GW-UserAgent 2973 4880 IN IP4 9.42.25.149
s=SIP Call
c=IN IP4 9.42.25.149
t=0 0
m=audio 16464 RTP/AVP 0 101
c=IN IP4 9.42.25.149
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=ptime:20
a=inactive
a=label:1
m=audio 16466 RTP/AVP 0 101
c=IN IP4 9.42.25.149
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=ptime:20
a=sendonly
a=label:2
m=video 16468 RTP/AVP 97
c=IN IP4 9.42.25.149
b=TIAS:1000000
a=rtpmap:97 H264/90000
a=fmtp:97 profile-level-id=42801E;packetization-mode=0
a=inactive
a=label:3
m=video 16470 RTP/AVP 97
c=IN IP4 9.42.25.149
b=TIAS:1000000
a=rtpmap:97 H264/90000
a=fmtp:97 profile-level-id=42801E;packetization-mode=0
a=sendonly
a=label:4

--uniqueBoundary
Content-Type: application/rs-metadata+xml
Content-Disposition: recording-session

<?xml version="1.0" encoding="UTF-8"?>
<recording xmlns="urn:ietf:params:xml:ns:recording:1">
…
    <stream stream_id="jIBTUf1BEeSAdKsYHx7YVg==" session_id="jH+2kf1BEeSAb6sYHx7YVg==">
        <label>1</label>
    </stream>
    <stream stream_id="jIBTUf1BEeSAdasYHx7YVg==" session_id="jH+2kf1BEeSAb6sYHx7YVg==">
        <label>3</label>
    </stream>
…
    <stream stream_id="jIBTUf1BEeSAd6sYHx7YVg==" session_id="jH+2kf1BEeSAb6sYHx7YVg==">
```

```
        <label>2</label>
    </stream>
    <stream stream_id="jIBTUf1BEeSAeKsYHx7YVg==" session_id="jH+2kf1BEeSAb6sYHx7YVg==">
        <label>4</label>
    </stream>
    <participantstreamassoc participant_id="jIBTUf1BEeSAc6sYHx7YVg==">
        <recv>jIBTUf1BEeSAd6sYHx7YVg==</recv>
        <recv>jIBTUf1BEeSAeKsYHx7YVg==</recv>
    </participantstreamassoc>
    <participantstreamassoc participant_id="jIBTUf1BEeSAdqsYHx7YVg==">
        <send>jIBTUf1BEeSAd6sYHx7YVg==</send>
        <send>jIBTUf1BEeSAeKsYHx7YVg==</send>
    </participantstreamassoc>
</recording>

--uniqueBoundary--
```


Example: Hold with Inactive Attribute in SDP

Here, you can see that video call is sent in the initial INVITE to recorder where both the participants send and receive audio and video streams. There are 2 audio and 2 video streams from both the participants each in the **`participantStream association`** element.

```
--uniqueBoundary
Content-Type: application/sdp
Content-Disposition: session;handling=required

v=0
o=CiscoSystemsSIP-GW-UserAgent 7476 1347 IN IP4 9.42.25.149
s=SIP Call
c=IN IP4 9.42.25.149
t=0 0
m=audio 16496 RTP/AVP 0 101
c=IN IP4 9.42.25.149
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=ptime:20
a=sendonly
a=label:1
m=audio 16498 RTP/AVP 0 101
c=IN IP4 9.42.25.149
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=ptime:20
a=sendonly
a=label:2
m=video 16500 RTP/AVP 97
c=IN IP4 9.42.25.149
b=TIAS:1000000
a=rtpmap:97 H264/90000
a=fmtp:97 profile-level-id=42801E;packetization-mode=0
a=sendonly
a=label:3
m=video 16502 RTP/AVP 97
c=IN IP4 9.42.25.149
b=TIAS:1000000
a=rtpmap:97 H264/90000
a=fmtp:97 profile-level-id=42801E;packetization-mode=0
a=sendonly
a=label:4
```

```
--uniqueBoundary
Content-Type: application/rs-metadata+xml
Content-Disposition: recording-session

<?xml version="1.0" encoding="UTF-8"?>
<recording xmlns="urn:ietf:params:xml:ns:recording:1">
…
    <stream stream_id="uV/B4f1BEeSAmKsYHx7YVg==" session_id="uV/B4f1BEeSAk6sYHx7YVg==">
        <label>1</label>
    </stream>
    <stream stream_id="uV/B4f1BEeSAmasYHx7YVg==" session_id="uV/B4f1BEeSAk6sYHx7YVg==">
        <label>3</label>
    </stream>
…
    <stream stream_id="uV/B4f1BEeSAm6sYHx7YVg==" session_id="uV/B4f1BEeSAk6sYHx7YVg==">
        <label>2</label>
    </stream>
    <stream stream_id="uV/B4f1BEeSAnKsYHx7YVg==" session_id="uV/B4f1BEeSAk6sYHx7YVg==">
        <label>4</label>
    </stream>
    <participantstreamassoc participant_id="uV/B4f1BEeSAl6sYHx7YVg==">
        <send>uV/B4f1BEeSAmKsYHx7YVg==</send>
        <recv>uV/B4f1BEeSAm6sYHx7YVg==</recv>
        <send>uV/B4f1BEeSAmasYHx7YVg==</send>
        <recv>uV/B4f1BEeSAnKsYHx7YVg==</recv>
    </participantstreamassoc>
    <participantstreamassoc participant_id="uV/B4f1BEeSAmqsYHx7YVg==">
        <send>uV/B4f1BEeSAm6sYHx7YVg==</send>
        <recv>uV/B4f1BEeSAmKsYHx7YVg==</recv>
        <send>uV/B4f1BEeSAnKsYHx7YVg==</send>
        <recv>uV/B4f1BEeSAmasYHx7YVg==</recv>
    </participantstreamassoc>
</recording>

--uniqueBoundary--
```

When the first participant puts the call on hold with inactive SDP attribute, there will be not any active streams in the metadata.

```
--uniqueBoundary
Content-Type: application/sdp

v=0
o=CiscoSystemsSIP-GW-UserAgent 7476 1348 IN IP4 9.42.25.149
s=SIP Call
c=IN IP4 9.42.25.149
t=0 0
m=audio 16496 RTP/AVP 0 101
c=IN IP4 9.42.25.149
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=ptime:20
a=inactive
a=label:1
m=audio 16498 RTP/AVP 0 101
c=IN IP4 9.42.25.149
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=ptime:20
a=inactive
a=label:2
m=video 16500 RTP/AVP 97
```

```
c=IN IP4 9.42.25.149
b=TIAS:1000000
a=rtpmap:97 H264/90000
a=fmtp:97 profile-level-id=42801E;packetization-mode=0
a=inactive
a=label:3
m=video 16502 RTP/AVP 97
c=IN IP4 9.42.25.149
b=TIAS:1000000
a=rtpmap:97 H264/90000
a=fmtp:97 profile-level-id=42801E;packetization-mode=0
a=inactive
a=label:4

--uniqueBoundary
Content-Type: application/rs-metadata+xml
Content-Disposition: recording-session

<?xml version="1.0" encoding="UTF-8"?>
<recording xmlns="urn:ietf:params:xml:ns:recording:1">
…
    <stream stream_id="uV/B4f1BEeSAmKsYHx7YVg==" session_id="uV/B4f1BEeSAk6sYHx7YVg==">
        <label>1</label>
    </stream>
    <stream stream_id="uV/B4f1BEeSAmasYHx7YVg==" session_id="uV/B4f1BEeSAk6sYHx7YVg==">
        <label>3</label>
    </stream>
…
    <stream stream_id="uV/B4f1BEeSAm6sYHx7YVg==" session_id="uV/B4f1BEeSAk6sYHx7YVg==">
        <label>2</label>
    </stream>
    <stream stream_id="uV/B4f1BEeSAnKsYHx7YVg==" session_id="uV/B4f1BEeSAk6sYHx7YVg==">
        <label>4</label>
    </stream>
    <participantstreamassoc participant_id="uV/B4f1BEeSAl6sYHx7YVg==">
    </participantstreamassoc>
    <participantstreamassoc participant_id="uV/B4f1BEeSAmqsYHx7YVg==">
    </participantstreamassoc>
</recording>

--uniqueBoundary--
```

Example: Escalation

During escalation, video streams will be added to the Re-INVITE meta-data sent to the recorder.

In the below example, you can see the metadata representation of an original audio call sent in the initial INVITE to the recorder where both the participants send and receive audio streams.

```
--uniqueBoundary
Content-Type: application/sdp
Content-Disposition: session;handling=required

v=0
o=CiscoSystemsSIP-GW-UserAgent 6360 4788 IN IP4 9.42.25.149
s=SIP Call
c=IN IP4 9.42.25.149
t=0 0
m=audio 16628 RTP/AVP 8 101
c=IN IP4 9.42.25.149
a=rtpmap:8 PCMA/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
```

```
a=ptime:20
a=sendonly
a=label:1
m=audio 16630 RTP/AVP 8 101
c=IN IP4 9.42.25.149
a=rtpmap:8 PCMA/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=ptime:20
a=sendonly
a=label:2

--uniqueBoundary
Content-Type: application/rs-metadata+xml
Content-Disposition: recording-session

<?xml version="1.0" encoding="UTF-8"?>
<recording xmlns="urn:ietf:params:xml:ns:recording:1">
…
    <stream stream_id="evyS5/1CEeSBOKsYHx7YVg==" session_id="evv2v/1CEeSBM6sYHx7YVg==">
        <label>1</label>
    </stream>
…
    <stream stream_id="evyS5/1CEeSBOqsYHx7YVg==" session_id="evv2v/1CEeSBM6sYHx7YVg==">
        <label>2</label>
    </stream>
    <participantstreamassoc participant_id="evyS5/1CEeSBN6sYHx7YVg==">
        <send>evyS5/1CEeSBOKsYHx7YVg==</send>
        <recv>evyS5/1CEeSBOqsYHx7YVg==</recv>
    </participantstreamassoc>
    <participantstreamassoc participant_id="evyS5/1CEeSBOasYHx7YVg==">
        <send>evyS5/1CEeSBOqsYHx7YVg==</send>
        <recv>evyS5/1CEeSBOKsYHx7YVg==</recv>
    </participantstreamassoc>
</recording>

--uniqueBoundary--
```

After escalation, video streams get added into the **`participantStream association`** element in metadata for both the participants. There will be 4 streams in total.

```
--uniqueBoundary
Content-Type: application/sdp

v=0
o=CiscoSystemsSIP-GW-UserAgent 6360 4789 IN IP4 9.42.25.149
s=SIP Call
c=IN IP4 9.42.25.149
t=0 0
m=audio 16628 RTP/AVP 18 101
c=IN IP4 9.42.25.149
a=rtpmap:18 G729/8000
a=fmtp:18 annexb=no
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=ptime:20
a=sendonly
a=label:1
m=audio 16630 RTP/AVP 18 101
c=IN IP4 9.42.25.149
a=rtpmap:18 G729/8000
a=fmtp:18 annexb=no
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=ptime:20
```

```
a=sendonly
a=label:2
m=video 16636 RTP/AVP 97
c=IN IP4 9.42.25.149
b=TIAS:1000000
a=rtpmap:97 H264/90000
a=fmtp:97 profile-level-id=42801E;packetization-mode=0
a=sendonly
a=label:3
m=video 16638 RTP/AVP 97
c=IN IP4 9.42.25.149
b=TIAS:1000000
a=rtpmap:97 H264/90000
a=fmtp:97 profile-level-id=42801E;packetization-mode=0
a=sendonly
a=label:4

--uniqueBoundary
Content-Type: application/rs-metadata+xml
Content-Disposition: recording-session

<?xml version="1.0" encoding="UTF-8"?>
<recording xmlns="urn:ietf:params:xml:ns:recording:1">
…
    <stream stream_id="evyS5/1CEeSBOKsYHx7YVg==" session_id="evv2v/1CEeSBM6sYHx7YVg==">
        <label>1</label>
    </stream>
    <stream stream_id="e5Zhtv1CEeSBPKsYHx7YVg==" session_id="evv2v/1CEeSBM6sYHx7YVg==">
        <label>3</label>
    </stream>
…
    <stream stream_id="e5Zhtv1CEeSBPasYHx7YVg==" session_id="evv2v/1CEeSBM6sYHx7YVg==">
        <label>4</label>
    </stream>
    <participantstreamassoc participant_id="evyS5/1CEeSBN6sYHx7YVg==">
        <send>evyS5/1CEeSBOKsYHx7YVg==</send>
        <recv>evyS5/1CEeSBOqsYHx7YVg==</recv>
        <send>e5Zhtv1CEeSBPKsYHx7YVg==</send>
        <recv>e5Zhtv1CEeSBPasYHx7YVg==</recv>
    </participantstreamassoc>
    <participantstreamassoc participant_id="evyS5/1CEeSBOasYHx7YVg==">
        <send>evyS5/1CEeSBOqsYHx7YVg==</send>
        <recv>evyS5/1CEeSBOKsYHx7YVg==</recv>
        <send>e5Zhtv1CEeSBPasYHx7YVg==</send>
        <recv>e5Zhtv1CEeSBPKsYHx7YVg==</recv>
    </participantstreamassoc>
</recording>

--uniqueBoundary--
```

Example: De-escalation

During de-escalation, video streams will be truncated in the Re-INVITE metadata sent to the recorder.

In the below example, you can see two streams each for the audio and video calls in the metadata.

```
--uniqueBoundary
Content-Type: application/sdp
Content-Disposition: session;handling=required

v=0
o=CiscoSystemsSIP-GW-UserAgent 7616 8308 IN IP4 9.42.25.149
s=SIP Call
```

```
c=IN IP4 9.42.25.149
t=0 0
m=audio 16648 RTP/AVP 116 101
c=IN IP4 9.42.25.149
a=rtpmap:116 iLBC/8000
a=fmtp:116 mode=20
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=ptime:20
a=maxptime:20
a=sendonly
a=label:1
m=audio 16650 RTP/AVP 116 101
c=IN IP4 9.42.25.149
a=rtpmap:116 iLBC/8000
a=fmtp:116 mode=20
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=ptime:20
a=maxptime:20
a=sendonly
a=label:2
m=video 16652 RTP/AVP 97
c=IN IP4 9.42.25.149
b=TIAS:1000000
a=rtpmap:97 H264/90000
a=fmtp:97 profile-level-id=42801E;packetization-mode=0
a=sendonly
a=label:3
m=video 16654 RTP/AVP 97
c=IN IP4 9.42.25.149
b=TIAS:1000000
a=rtpmap:97 H264/90000
a=fmtp:97 profile-level-id=42801E;packetization-mode=0
a=sendonly
a=label:4

--uniqueBoundary
Content-Type: application/rs-metadata+xml
Content-Disposition: recording-session

<?xml version="1.0" encoding="UTF-8"?>
<recording xmlns="urn:ietf:params:xml:ns:recording:1">
…
    <stream stream_id="j5OQdP1CEeSBSqsYHx7YVg==" session_id="j5L0TP1CEeSBRasYHx7YVg==">
        <label>1</label>
    </stream>
    <stream stream_id="j5OQdP1CEeSBS6sYHx7YVg==" session_id="j5L0TP1CEeSBRasYHx7YVg==">
        <label>3</label>
    </stream>
…
    <stream stream_id="j5OQdP1CEeSBTasYHx7YVg==" session_id="j5L0TP1CEeSBRasYHx7YVg==">
        <label>2</label>
    </stream>
    <stream stream_id="j5OQdP1CEeSBTqsYHx7YVg==" session_id="j5L0TP1CEeSBRasYHx7YVg==">
        <label>4</label>
    </stream>
    <participantstreamassoc participant_id="j5OQdP1CEeSBSasYHx7YVg==">
        <send>j5OQdP1CEeSBSqsYHx7YVg==</send>
        <recv>j5OQdP1CEeSBTasYHx7YVg==</recv>
        <send>j5OQdP1CEeSBS6sYHx7YVg==</send>
        <recv>j5OQdP1CEeSBTqsYHx7YVg==</recv>
    </participantstreamassoc>
    <participantstreamassoc participant_id="j5OQdP1CEeSBTKsYHx7YVg==">
```

```
        <send>j5OQdP1CEeSBTasYHx7YVg==</send>
        <recv>j5OQdP1CEeSBSqsYHx7YVg==</recv>
        <send>j5OQdP1CEeSBTqsYHx7YVg==</send>
        <recv>j5OQdP1CEeSBS6sYHx7YVg==</recv>
    </participantstreamassoc>
</recording>

--uniqueBoundary--
```

After de-escalation, video streams are removed from the metadata and only audio calls will be present in the **`participantStream association`** element.

```
--uniqueBoundary
Content-Type: application/sdp

v=0
o=CiscoSystemsSIP-GW-UserAgent 7616 8309 IN IP4 9.42.25.149
s=SIP Call
c=IN IP4 9.42.25.149
t=0 0
m=audio 16648 RTP/AVP 0 101
c=IN IP4 9.42.25.149
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=ptime:20
a=sendonly
a=label:1
m=audio 16650 RTP/AVP 0 101
c=IN IP4 9.42.25.149
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=ptime:20
a=sendonly
a=label:2
m=video 0 RTP/AVP 97
c=IN IP4 9.42.25.149
b=TIAS:1000000
a=rtpmap:97 H264/90000
a=fmtp:97 profile-level-id=42801E;packetization-mode=0
a=sendonly
a=label:3
m=video 0 RTP/AVP 97
c=IN IP4 9.42.25.149
b=TIAS:1000000
a=rtpmap:97 H264/90000
a=fmtp:97 profile-level-id=42801E;packetization-mode=0
a=sendonly
a=label:4

--uniqueBoundary
Content-Type: application/rs-metadata+xml
Content-Disposition: recording-session

<?xml version="1.0" encoding="UTF-8"?>
<recording xmlns="urn:ietf:params:xml:ns:recording:1">
…
    <stream stream_id="j5OQdP1CEeSBSqsYHx7YVg==" session_id="j5L0TP1CEeSBRasYHx7YVg==">
        <label>1</label>
    </stream>
…
    <stream stream_id="j5OQdP1CEeSBTasYHx7YVg==" session_id="j5L0TP1CEeSBRasYHx7YVg==">
        <label>2</label>
    </stream>
```

```
    <participantstreamassoc participant_id="j5OQdP1CEeSBSasYHx7YVg==">
        <send>j5OQdP1CEeSBSqsYHx7YVg==</send>
        <recv>j5OQdP1CEeSBTasYHx7YVg==</recv>
    </participantstreamassoc>
    <participantstreamassoc participant_id="j5OQdP1CEeSBTKsYHx7YVg==">
        <send>j5OQdP1CEeSBTasYHx7YVg==</send>
        <recv>j5OQdP1CEeSBSqsYHx7YVg==</recv>
    </participantstreamassoc>
</recording>

--uniqueBoundary--
```

Configuration Example for Metadata Variations with Different Transfer Flows

Example: Transfer of Re-INVITE/REFER Consume Scenario

In the case of Re-INVITE or REFER Consume transfer scenarios, CUBE receives re-INVITE with caller-id change. This re-INVITE will have the remote-party-ID details.

After transfer, participant A is disassociated from the call and participant C joins the call. This information is provided in the metadata sent to the recording server. Here, **7774442214** associates and **7774442212** disassociates from the call.

```
INVITE sip:7774442216@10.64.86.102:5060;transport=tcp SIP/2.0
From: <sip:7774442212@10.104.54.52>;tag=498652~97a89a01
To: <sip:7774442216@10.64.86.102>;tag=7C798-1441
...
...
Remote-Party-ID: <sip:7774442214@10.104.54.52>;party=calling;screen=yes;privacy=off
Contact: <sip:7774442214@10.104.54.52:5060;transport=tcp>
...
 <participant participant_id="vm+z2xM6EeWAIN4iOrLrag==">
        <nameID aor="sip:7774442214@10.104.54.52">
        </nameID>
    </participant>
    <participantsessionassoc participant_id="vm+z2xM6EeWAIN4iOrLrag=="
session_id="vACJ+xM6EeWAF94iOrLrag==">
        <associate-time>2015-06-16T08:44:32.869Z</associate-time>
    </participantsessionassoc>
...
 <participant participant_id="vACJ+xM6EeWAGN4iOrLrag==">
        <nameID aor="sip:7774442212@10.104.54.52">
        </nameID>
    </participant>
    <participantsessionassoc participant_id="vACJ+xM6EeWAGN4iOrLrag=="
session_id="vACJ+xM6EeWAGN4iOrLrag==">
        <disassociate-time>2015-06-16T08:44:32.869Z</disassociate-time>
</participantsessionassoc>
...
```

Configuaration Examples for Metadata Variations with Caller-ID UPDATE Flow

Example: Caller-ID UPDATE Request and Response Scenario

In case of Re-INVITE based transfer, any UPDATE request will contain caller-id changes. These changes are forwarded to the remote party and once CUBE receives a 200OK message, the remote-party-ID details are transferred.

The response of UPDATE request contains the associated caller-id changes. The CUBE forwards the response UPDATE information to the remote party with caller-id changes after the UPDATE request. From the metadata, you can see that the participants A and C disassociate from the call and participants B and D joins (associates) the call. Here, **7774442212** and **7774442216** disassociates from the call and **7774442214** and **7774442218** joins the call after the caller-id update.

```
UPDATE sip:7774442216@10.64.86.102:5060;transport=tcp SIP/2.0
From: <sip:7774442212@10.104.54.52>;tag=498652~97a89a01
To: <sip:7774442216@10.64.86.102>;tag=7C798-1441
...
...
Remote-Party-ID: <sip:7774442214@10.104.54.52>;party=calling;screen=yes;privacy=off
Contact: <sip:7774442214@10.104.54.52:5060;transport=tcp>

Response of UPDATE contains caller-id changes
...
SIP/2.0 200 OK
From: <sip:7774442212@10.64.86.102>;tag=7C78C-1E7C
To: <sip:7774442216@10.104.54.52>;tag=498653~97a89a01
...
Remote-Party-ID: <sip:7774442218@10.104.54.52>;party=called;screen=yes;privacy=off
Contact: <sip:7774442218@10.104.54.52:5060>
Content-Length: 0

...
    <participant participant_id="vm+z2xM6EeWAIN4iOrLrag==">
        <nameID aor="sip:7774442214@10.104.54.52">
        </nameID>
    </participant>
    <participantsessionassoc participant_id="vm+z2xM6EeWAIN4iOrLrag=="
session_id="vACJ+xM6EeWAF94iOrLrag==">
        <associate-time>2015-06-16T08:44:32.869Z</associate-time>
    </participantsessionassoc>
    <participant participant_id="vm+z2xM6EeWAIN4iOrLrag==">
        <nameID aor="sip:7774442218@10.104.54.52">
        </nameID>
    </participant>
    <participantsessionassoc participant_id="vm+z2xM6EeWAIN4iOrLrag=="
session_id="vACJ+xM6EeWAF94iOrLrag==">
        <associate-time>2015-06-16T08:44:32.869Z</associate-time>
    </participantsessionassoc>
    <participant participant_id="vACJ+xM6EeWAGN4iOrLrag==">
        <nameID aor="sip:7774442212@10.104.54.52">
        </nameID>
    </participant>
    <participantsessionassoc participant_id="vACJ+xM6EeWAGN4iOrLrag=="
session_id="vACJ+xM6EeWAGN4iOrLrag==">
        <disassociate-time>2015-06-16T08:44:32.869Z</disassociate-time>
```

```
    </participantsessionassoc>
    <participant participant_id="vACJ+xM6EeWAGN4iOrLrag==">
        <nameID aor="sip:7774442216@10.104.54.52">
        </nameID>
    </participant>
    <participantsessionassoc participant_id="vACJ+xM6EeWAGN4iOrLrag=="
session_id="vACJ+xM6EeWAGN4iOrLrag==">
        <disassociate-time>2015-06-16T08:44:32.869Z</disassociate-time>
    </participantsessionassoc>
...
```

Configuration Example for Metadata Variations with Call Disconnect

Example: Disconnect while Sending Metadata with BYE

When the original call disconnects without any reason, CUBE initiates a BYE session with the recording server along with the metadata.

In this case, the metadata contains the end time of the session along with the disassociation time of all the active participants from the call.

```
BYE sip:5555555@8.41.17.71:13961;transport=UDP SIP/2.0
...
Reason: Q.850;cause=16
Content-Type: application/rs-metadata+xml
Content-Disposition: recording-session
Content-Length: 984

<?xml version="1.0" encoding="UTF-8"?>
<recording xmlns="urn:ietf:params:xml:ns:recording:1">
    <datamode>complete</datamode>
    <session session_id="t5nW8RM6EeWACN4iOrLrag==">
        <end-time>2015-06-16T08:44:36.661Z</end-time>
    </session>
    <participant participant_id="t5nW8RM6EeWACt4iOrLrag==">
        <nameID aor="sip:7774442212@10.104.54.52">
        </nameID>
    </participant>
    <participantsessionassoc participant_id="t5nW8RM6EeWACt4iOrLrag=="
session_id="t5nW8RM6EeWACt4iOrLrag==">
        <disassociate-time>2015-06-16T08:44:36.657Z</disassociate-time>
    </participantsessionassoc>
    <participant participant_id="t5nW8RM6EeWACd4iOrLrag==">
        <nameID aor="sip:7774442214@10.104.54.52">
        </nameID>
    </participant>
    <participantsessionassoc participant_id="t5nW8RM6EeWACd4iOrLrag=="
session_id="t5nW8RM6EeWACd4iOrLrag==">
        <disassociate-time>2015-06-16T08:44:36.657Z</disassociate-time>
    </participantsessionassoc>
</recording>
```