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, on page 1
- Prerequisites for SIPREC Recording, on page 2
- Restrictions for SIPREC Recording, on page 2
- Information About SIPREC Recording Using CUBE, on page 3
- How to Configure SIPREC-Based Recording, on page 4
- Configuration Examples for SIPREC-based Recording, on page 9
- Example of Metadata Variations with Different Mid-call Flows, on page 9
- Example of Metadata Variations with Different Transfer Flows, on page 22
- Example of Metadata Variations with Caller-ID UPDATE Flow, on page 22
- Example of Metadata Variations with Call Disconnect, on page 24

### 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 www.cisco.com/go/cfn. 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>SIPREC (SIP Recording)</td>
<td>Cisco IOS 15.6(1)T</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>Cisco IOS XE 3.17S</td>
<td></td>
</tr>
</tbody>
</table>

Feature Name

- SIPREC (SIP Recording)

Releases

- Cisco IOS 15.6(1)T
- Cisco IOS XE 3.17S

Feature Information

- 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

- 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 on Network-Based Recording, see http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/voice/cube/configuration/cube-book/voi-ntwk-based.html.
- 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
- Media mixing on forked streams is not supported
- Digital Signal Processing (DSP) resources are not supported on forked legs

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

• 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 represents 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 tag siprec
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
### 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 |
| **Step 2** configure terminal | Enters global configuration mode. |
  Example: Device# configure terminal |
| **Step 3** media profile recorder *profile-tag* | Configures the media profile recorder and enters media profile configuration mode.  
  Example: Device(config)# media profile recorder 100 |
| **Step 4** (Optional) 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.  
  Example: Device(cfg-mediaprofile)# media-type audio |
| **Step 5** media-recording *dial-peer-tag*  
  [dial-peer-tag2...dial-peer-tag5] | Configures the dial-peers that need to be configured  
  Note You can specify a maximum of five dial-peer tags.  
  Example: Device(cfg-mediaprofile)# media-recording 8000 8001 8002 |
| **Step 6** exit | Exits media profile configuration mode.  
  Example: Device(cfg-mediaprofile)# exit |
| **Step 7** media class *tag* | Configures a media class and enters media class configuration mode.  
  Example: Device(config)# media class 100 |
| **Step 8** recorder profile *tag siprec* | Configures the media profile SIPREC recorder.  
  Example: Device(cfg-mediaclass)# recorder profile 201 siprec |
<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td><code>exit</code></td>
<td>Exits media class configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config)# exit</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td><code>dial-peer voice dummy-recorder-dial-peer-tag voip</code></td>
<td>Configures a recorder dial peer and enters dial peer voice configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config)# dial-peer voice 8000 voip</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td><code>media-class tag</code></td>
<td>Configures media class on a dial peer.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config-dial-peer)# media-class 100</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td><code>destination-pattern [+] string [T]</code></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.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config-dial-peer)# destination-pattern 595959</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td><code>session protocol sipv2</code></td>
<td>Configures the VoIP dial peer to use Session Initiation Protocol (SIP).</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config-dial-peer)# session protocol sipv2</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>`session target ipv4:[recording-server-destination-address</td>
<td>recording-server-dns]`</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config-dial-peer)# session target ipv4:10.42.29.7</td>
<td>in this format: xxx.xxx.xxx.xxx</td>
</tr>
<tr>
<td>15</td>
<td><code>session transport tcp</code></td>
<td>Configures a VoIP dial peer to use Transmission Control Protocol (TCP).</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config-dial-peer)# session transport tcp</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td><code>end</code></td>
<td>Returns to privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config-dial-peer)# end</td>
<td></td>
</tr>
</tbody>
</table>
Configuring SIPREC-Based Recording (without Media Profile Recorder)

**SUMMARY STEPS**

1. enable
2. configure terminal
3. media class tag
4. recorder parameters siprec
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>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></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>media class tag</td>
<td>Configures the media class and enters media class configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(config)# media class 100</td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>recorder parameters siprec</td>
<td>Enables SIPREC recording.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Device(cfg-mediaclass)# recorder parameter siprec</td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td>(Optional) media-type audio</td>
<td>Configures recording of audio only in a call with both audio and video.</td>
</tr>
<tr>
<td></td>
<td>Example:</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><strong>media-recording dial-peer-tag</strong>&lt;br&gt;Example: &lt;br&gt;Device(cfg-mediaclass-recorder)# media-recording 8000, 8001, 8002</td>
<td>Configures voice-class recording parameters.&lt;br&gt;Note You can specify a maximum of five dial-peer tags.</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td><strong>exit</strong>&lt;br&gt;Example: &lt;br&gt;Device(cfg-mediaclass-recorder)# exit</td>
<td>Exits media class recorder parameter configuration mode.</td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td><strong>exit</strong>&lt;br&gt;Example: &lt;br&gt;Device(cfg-mediaclass)# exit</td>
<td>Exits media class configuration mode.</td>
</tr>
<tr>
<td><strong>Step 9</strong></td>
<td><strong>dial-peer voice dummy-recorder-dial-peer-tag voip</strong>&lt;br&gt;Example: &lt;br&gt;Device(config)# dial-peer voice 8000 voip</td>
<td>Configures a recorder dial peer and enters dial peer voice configuration mode.</td>
</tr>
<tr>
<td><strong>Step 10</strong></td>
<td><strong>media-class tag</strong>&lt;br&gt;Example: &lt;br&gt;Device(config-dial-peer)# media-class 100</td>
<td>Configures media class on a dial peer.</td>
</tr>
<tr>
<td><strong>Step 11</strong></td>
<td><strong>destination-pattern [+] string [T]</strong>&lt;br&gt;Example: &lt;br&gt;Device(config-dial-peer)# destination-pattern 595959</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><strong>Step 12</strong></td>
<td><strong>session protocol sipv2</strong>&lt;br&gt;Example: &lt;br&gt;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 13</strong></td>
<td><strong>session target ipv4:</strong>[recording-server-destination-address</td>
<td>recording-server-dns]&lt;br&gt;Example: &lt;br&gt;Device(config-dial-peer)# session target ipv4:10.42.29.7</td>
</tr>
</tbody>
</table>

---

**Purpose Command or Action**

**Device(cfg-mediaprofile)# media-type audio**

If this configuration is not done, both audio and video are recorded.

**Note**

Configures voice-class recording parameters.

**Step 6**

**media-recording dial-peer-tag**

**Example:**

Device(cfg-mediaclass-recorder)# media-recording 8000, 8001, 8002

**Step 7**

**exit**

**Example:**

Device(cfg-mediaclass-recorder)# exit

**Step 8**

**exit**

**Example:**

Device(cfg-mediaclass)# exit

**Step 9**

**dial-peer voice dummy-recorder-dial-peer-tag voip**

**Example:**

Device(config)# dial-peer voice 8000 voip

**Step 10**

**media-class tag**

**Example:**

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

**Step 11**

**destination-pattern [+] string [T]**

**Example:**

Device(config-dial-peer)# destination-pattern 595959

**Step 12**

**session protocol sipv2**

**Example:**

Device(config-dial-peer)# session protocol sipv2

**Step 13**

**session target ipv4:**[recording-server-destination-address | recording-server-dns]

**Example:**

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

---
### Purpose

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 14</strong></td>
<td><strong>Purpose</strong></td>
</tr>
<tr>
<td>session transport tcp</td>
<td>Configures a VoIP dial peer to use Transmission Control Protocol (TCP).</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device(config-dial-peer)# session transport tcp</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Step 15</strong></th>
<th><strong>Purpose</strong></th>
</tr>
</thead>
<tbody>
<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>

### 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
```

### Example of 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
c=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
```

**SIPREC (SIP Recording)**

**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
```

**Example of 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
c=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
```
Example: Complete SIP Recording Metadata information sent in INVITE or Re-INVITE

SIPREC (SIP Recording)

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="JaPQE1CEeSA6sYHx7YVg==">
    <start-time>2015-05-19T09:42:06.911Z</start-time>
  </session>
  <participant participant_id="JaPQE1CEeSA76sYHx7YVg==">
    <nameID aor="sip:808808@9.0.0.174">
      <name xml:lang="en">808808</name>
    </nameID>
  </participant>
  <participantsessionassoc participant_id="JaPQE1CEeSA76sYHx7YVg==" session_id="JaPQE1CEeSA6sYHx7YVg==">
    <associate-time>2015-05-19T09:42:06.911Z</associate-time>
  </participantsessionassoc>
  <stream stream_id="JaPQE1CEeSA8sYHx7YVg==" session_id="JaPQE1CEeSA6sYHx7YVg==">
    <label>1</label>
  </stream>
  <stream stream_id="JaPQE1CEeSA8asYHx7YVg==" session_id="JaPQE1CEeSA6sYHx7YVg==">
    <label>3</label>
  </stream>
  <participant participant_id="JaPQE1CEeSA8qsYHx7YVg==">
    <nameID aor="sip:909909@9.0.0.174">
      <name xml:lang="en">909909</name>
    </nameID>
  </participant>
  <participantsessionassoc participant_id="JaPQE1CEeSA8qsYHx7YVg==" session_id="JaPQE1CEeSA6sYHx7YVg==">
    <associate-time>2015-05-19T09:42:06.911Z</associate-time>
  </participantsessionassoc>
  <stream stream_id="JaPQE1CEeSA86sYHx7YVg==" session_id="JaPQE1CEeSA6sYHx7YVg==">
    <label>2</label>
  </stream>
</recording>
Example: Complete SIP Recording Metadata information sent in INVITE or Re-INVITE

```xml
<stream stream_id="JaPqePlCEeSA9KsYHx7YVg==" session_id="JaPqePlCEeSA66sYHx7YVg==">
  <label>4</label>
</stream>

<participantstreamassoc participant_id="JaPqePlCEeSA76sYHx7YVg==">
  <send>JaPqePlCEeSA88sYHx7YVg==</send>
  <recv>JaPqePlCEeSA866sYHx7YVg==</recv>
</participantstreamassoc>

<participantstreamassoc participant_id="JaPqePlCEeSA8qsYHx7YVg==">
  <send>JaPqePlCEeSA868sYHx7YVg==</send>
  <recv>JaPqePlCEeSA886sYHx7YVg==</recv>
</participantstreamassoc>

<participantstreamassoc participant_id="JaPqePlCEeSA9KsYHx7YVg==">
  <send>JaPqePlCEeSA98asYHx7YVg==</send>
  <recv>JaPqePlCEeSA99asYHx7YVg==</recv>
</participantstreamassoc>

<participantstreamassoc participant_id="JaPqePlCEeSA8asYHx7YVg==">
  <send>JaPqePlCEeSA89asYHx7YVg==</send>
  <recv>JaPqePlCEeSA98asYHx7YVg==</recv>
</participantstreamassoc>
</recording>

---uniqueBoundary---

<table>
<thead>
<tr>
<th>Output Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dataMode&gt;complete&lt;/dataMode</td>
<td>&lt;dataMode&gt; is a recording element that indicates whether the XML document is a complete document or a partial update. If no &lt;dataMode&gt; element is present then the default value is &quot;complete&quot;.</td>
</tr>
<tr>
<td>session_id=&quot;JaPqePlCEeSA66sYHx7YVg==&quot;</td>
<td>Session ID which remains constant for the complete call leg.</td>
</tr>
<tr>
<td>nameID aor=&quot;sip:808808@9.0.0.174&quot;</td>
<td>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.</td>
</tr>
<tr>
<td>a=label:1;</td>
<td>The &lt;stream&gt; element represents a Media Stream object. Stream element indicates the SDP media lines associated with the session and participants.</td>
</tr>
<tr>
<td>session_id=&quot;JaPqePlCEeSA66sYHx7YVg==&quot;</td>
<td>The &lt;label&gt; element within the &lt;stream&gt; element references an SDP &quot;a=label&quot; attribute that identifies an m-line within the RS SDP. This m-line carries the media stream from the SRC to the SRS.</td>
</tr>
</tbody>
</table>
### 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.

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

v=0
c=IN IP4 9.42.25.149
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
```
Example: Hold with Send-only / Recv-only Attribute in SDP

```
a=fmtp:101 0-16
a=rtime: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=rtime: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="jIBTUf1BfEeSAdKsYHx7YVg==" session_id="jH+2kf1BEeSAb6sYHx7YVg==">
   <label>1</label>
</stream>
...<stream stream_id="jIBTUf1BfEeSAd6sYHx7YVg==" session_id="jH+2kf1BEeSAb6sYHx7YVg==">
   <label>3</label>
</stream>
...<stream stream_id="jIBTUf1BfEeSAd6sYHx7YVg==" session_id="jH+2kf1BEeSAb6sYHx7YVg==">
   <label>2</label>
</stream>
...<stream stream_id="jIBTUf1BfEeSAd6sYHx7YVg==" session_id="jH+2kf1BEeSAb6sYHx7YVg==">
   <label>4</label>
</stream>

<participantstreamassoc participant_id="jIBTUf1BfEeSAC6sYHx7YVg==">
   <send>jIBTUf1BfEeSAdKsYHx7YVg==</send>
   <recv>jIBTUf1BfEeSAd6sYHx7YVg==</recv>
   <send>jIBTUf1BfEeSAdasYHx7YVg==</send>
   <recv>jIBTUf1BfEeSaeKsYHx7YVg==</recv>
</participantstreamassoc>

<participantstreamassoc participant_id="jIBTUf1BfEeSAdqsYHx7YVg==">
   <send>jIBTUf1BfEeSAd6sYHx7YVg==</send>
   <recv>jIBTUf1BfEeSAdKsYHx7YVg==</recv>
   <send>jIBTUf1BfEeSAdasYHx7YVg==</send>
   <recv>jIBTUf1BfEeSaeKsYHx7YVg==</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:

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

v=0
c=IN IP4 9.42.25.149
m=audio 16464 RTP/AVP 0 101
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=inactive
m=audio 16466 RTP/AVP 0 101
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
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
m=video 16470 RTP/AVP 97
m=video 16472 RTP/AVP 97
--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="jIBTUf1BEeSAdR6sYHx77YVg==" session_id="jH+2kf1BEeSAb6sYHx77YVg==">
    <label>1</label>
  </stream>
  <stream stream_id="jIBTUf1BEeSAdasYHx77YVg==" session_id="jH+2kf1BEeSAb6sYHx77YVg==">
    <label>3</label>
  </stream>
  ...
  <stream stream_id="jIBTUf1BEeSAd6sYHx77YVg==" session_id="jH+2kf1BEeSAb6sYHx77YVg==">
    <label>2</label>
  </stream>
  <stream stream_id="jIBTUf1BEeSAsY7YVg==" session_id="jH+2kf1BEeSAb6sYHx77YVg==">
    <label>4</label>
  </stream>
</recording>
```
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.

```
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
```

```xml
--uniqueBoundary
Content-Type: application/rs-metadata+xml
Content-Disposition: recording-session
```
When the first participant puts the call on hold with inactive SDP attribute, there will be not any active streams in the metadata.
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.

```
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=sendonly
a=label:1
m=audio 16630 RTP/AVP 8 101
```
After escalation, video streams get added into the **participantStream association** element in metadata for both the participants. There will be 4 streams in total.
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.

```xml
<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
```
Example: De-escalation
After de-escalation, video streams are removed from the metadata and only audio calls will be present in the participantStream association element.
Example of 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>
... 

Example of 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@10.104.54.52>;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
...

Example: Caller-ID UPDATE Request and Response Scenario
Example of 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>