



# Video Recording

---

- [Overview, on page 1](#)
- [Configure Video Forking, on page 2](#)
- [Verify for Video Forking, on page 5](#)

## Overview

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

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](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

**Table 1: Feature Information for Network-Based Recording of Video Calls Using Cisco Unified Border Element**

Feature Name	Releases	Feature Information
Network-Based Recording of Video Calls Using Cisco Unified Border Element	Baseline Functionality	The following commands were introduced or modified: <b>media profile video</b> , <b>ref-frame-req rtp</b> , <b>ref-frame-req sip-info</b> , <b>video profile</b> , <b>h264-packetization-mode</b> , <b>monitor-ref-frames</b> .

## 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).

## Configure Video Forking

### Enabling FIR for Video Calls (Using RTCP or 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**

	<b>Command or Action</b>	<b>Purpose</b>
<b>Step 1</b>	<b>enable</b>  <b>Example:</b> Device> enable	Enables privileged EXEC mode.
<b>Step 2</b>	<b>configure terminal</b>  <b>Example:</b> Device# configure terminal	Enters global configuration mode.
<b>Step 3</b>	<b>media profile video media-profile-tag</b>  <b>Example:</b> Device(config)# media profile video 1	Configures a video media profile and enters media profile configuration mode.
<b>Step 4</b>	Do one of the following:  <ul style="list-style-type: none"> <li>• <b>ref-frame-req rtpc retransmit-count</b> <i>retransmit-number</i></li> <li>• <b>ref-frame-req sip-info</b></li> </ul> <b>Example:</b> Device(cfg-mediaprofile)# ref-frame-req rtpc retransmit-count 4  <b>Example:</b> Device(cfg-mediaprofile)# ref-frame-req sip-info	Enables FIR using the RTCP or SIP INFO method.
<b>Step 5</b>	<b>end</b>  <b>Example:</b> Device(cfg-mediaprofile)# end	Exits media profile configuration mode.

**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**

	<b>Command or Action</b>	<b>Purpose</b>
<b>Step 1</b>	<b>enable</b>  <b>Example:</b> Device> enable	Enables privileged EXEC mode.
<b>Step 2</b>	<b>configure terminal</b>  <b>Example:</b> Device# configure terminal	Enters global configuration mode.
<b>Step 3</b>	<b>media profile video media-profile-tag</b>  <b>Example:</b> Device(config)# media profile video 1	Configures a video media profile and enters media profile configuration mode.
<b>Step 4</b>	<b>h264-packetization-mode packetization mode</b>  <b>Example:</b> Device(cfg-mediaprofile)# h264-packetization-mode 2	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.
<b>Step 5</b>	<b>end</b>  <b>Example:</b> Device(cfg-mediaprofile)# end	Exits media profile configuration mode.

**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 IDRs.

**SUMMARY STEPS**

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

**DETAILED STEPS**

	<b>Command or Action</b>	<b>Purpose</b>
<b>Step 1</b>	<b>enable</b>  <b>Example:</b> Device> enable	Enables privileged EXEC mode.

	<b>Command or Action</b>	<b>Purpose</b>
<b>Step 2</b>	<b>configure terminal</b>  <b>Example:</b> Device# configure terminal	Enters global configuration mode.
<b>Step 3</b>	<b>media profile video media-profile-tag</b>  <b>Example:</b> Device(config)# media profile video 1	Configures a video media profile and enters media profile configuration mode.
<b>Step 4</b>	<b>monitor-ref-frames</b>  <b>Example:</b> Device(cfg-mediaprofile)# monitor-ref-frames	Monitors reference frames or intra-frames.
<b>Step 5</b>	<b>end</b>  <b>Example:</b> Device(cfg-mediaprofile)# end	Exits media profile configuration mode.

## Verify for Video Forking

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
```

**Verify for Video Forking**

```
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=1

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

**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.

---