



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Active Recording Ecosystem Solution Guide

Media Server Recorder

12/19/2025

Media Server Recorder

A recording session is basically a SIP session initiated by Media Server (routed via GVP Resource Manager) to the recorder. The following subsections describe the behavior of the recording session.

Format

Format of Request URI

A simple SIP INVITE request should be used for sending a request from Media Server to the recorder to enable recording from a replicated media stream. The request URI contains:

1. Call UUID
2. Unique identifier for the recording session, this can be generated by Media Server or supplied in the `<param name="id"/>`
3. The DN of the party being recorded
4. The DN of the party that initiated the recording
5. Other parameters for the recording session

Depending on the recorder type, Media Server can be configured to send one SIP recording session per party in the call or a single SIP recording session containing both parties in the call.

For example, DN1 and DN2 are being recorded in the call and DN1 is the party that initiated the recording.

For recorders that accept one recording session per party, the following shows the requests URIs for each recording session:

```
sip:record=unique-  
identifier@recorder_address;calluuid=Call_UUID;dn=DN1;recordDN=DN1;other_parameters...
```

```
sip:record=unique-  
identifier@recorder_address;calluuid=Call_UUID;dn=DN2;recordDN=DN1;other_parameters...
```

The following shows the request URI when Media Server establishes a single recording session for both parties:

```
sip:record=unique-  
identifier@recorder_address;calluuid=Call_UUID;dn=DN1;otherDN=DN2;recordDN=DN...
```

Mapping

Mapping SDP Media Streams

One recording session per party

In the initial INVITE from Media Server, Media Server must provide the SDP offer or answer copied from party A or party B. Since the SDP answer provides the list of codecs being selected, Media Server should take the codec list and pass it as the m= line. If video media stream is also included by party A and B, by default Media Server must offer both media types (two m= lines) in the recording session unless the recording request specifies otherwise.

The recorder must generate an SDP answer in the 200 OK response containing the IP and port in the answer. Media server must take this IP:port as the destination for the replicated media stream. The rest of the SDP content is irrelevant to Media Server.

The recorder may reject recording of video media stream by setting to zero port in the SDP answer. Media server must allow the recording session to continue but also indicate video recording state as off.

One recording session for both parties

If only audio media type is recorded, then the SDP offer will contain only two media m= lines. The first m= line must be the media of the party identified by dn=<DN>, and the second m= line must be the media of party identified by otherDN=<DN2>.

Mid-dialog

Mid-dialog Session Updates

There are a few scenarios where the recording session needs to be updated:

- **Pause**—Whenever pause is requested by TrequestPrivateService, Media Server will send a reINVITE request to the recorder and set the SDP with a=inactive for all media streams to pause the streams.
- **Resume**—Whenever resume is requested by TrequestPrivateService, Media Server will send a reINVITE request to the recorder and set the SDP with a=sendonly for all media streams to resume the streams.
- **Mute**—Whenever one of the party in the communication session updates the SDP to mute the line, Media Server will send a reINVITE to the recorder and set the SDP with a=inactive for all media streams to pause the streams.
- **Call transfer/conference/consultation**—Whenever there is a change to change/add/remove party in the call, Media Server will send a reINVITE request to the recorder to update the SDP. All of the session attributes and SIP dialog headers remain unchanged except the version in the o= line is incremented.
- **Session timer**—As per RFC4028, Media Server will send a reINVITE with no SDP changes.

Secured Recording

Secured Recording Session

When the recording session requires to be secured, there are a few scenarios to consider:

- If the communication session is secured with SIPS/SRTP, Media Server will receive the encrypted media stream that Media Server will proxy between the endpoints. However, Media Server shall not pass the same encrypted media stream to the recording session and provide the existing crypto keys to the recording server. This is apparently a violation of security for a middle man to escrow security keys to another party. The proper way to establish a secure recording session is by decrypting the communication session and re-encrypt the recording session using a new set of keys negotiated between recording client and recording server. This adds performance burden to Media Server to process decryption and encryption.
- If the communication session is not secured but the recording session is required to be secured, Media Server will need to encrypt the recorded media towards the recording server.
- If the communication session is secure by the recording session does not require security, then Media Server will need to decrypt the recorded media towards the recording server.

Failures

Failure Handling

The Recording Session may fail to establish for various reasons:

- Recorder responds negatively to the INVITE request of the recording session
- Resource Manager fails to find an available recorder instance due to failures
- Resource Manager runs out of Recorder resources

As soon as Media Server recognizes the recording session has failed, Media Server notifies SIP Server that the recording is stopped. SIP Server updates the recording indication `GSIP_RECORD` to `off`, so the agent will be aware that the recording is not active.