

SIPREC	L.P. Portman, Ed.
Internet-Draft	NICE Systems
Intended status: Standards Track	H. Lum, Ed.
Expires: May 03, 2012	Genesys, Alcatel-Lucent
	A. Johnston
	Avaya
	A. Hutton
	Siemens Enterprise Communications
	October 31, 2011

Session Recording Protocol  
draft-ietf-siprec-protocol-02

## [Abstract](#)

This document specifies the use of the Session Initiation Protocol (SIP), the Session Description Protocol (SDP), and the Real Time Protocol (RTP) for delivering real-time media and metadata from a Communication Session (CS) to a recording device. The Session Recording Protocol specifies the use of SIP, SDP, and RTP to establish a Recording Session (RS) from the Session Recording Client (SRC), which is on the path of the CS, to a Session Recording Server (SRS) at the recording device.

## [Status of this Memo](#)

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet- Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on May 03, 2012.

## [Copyright Notice](#)

Copyright (c) 2011 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted

from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

- \*1. [Introduction](#)
- \*2. [Definitions](#)
- \*3. [Scope](#)
- \*4. [Overview of operations](#)
  - \*4.1. [Delivering recorded media](#)
  - \*4.2. [Delivering recording metadata](#)
- \*5. [Initiating a Recording Session](#)
  - \*5.1. [Procedures at the SRC](#)
  - \*5.2. [Procedures at the SRS](#)
- \*6. [SDP Handling](#)
  - \*6.1. [Procedures at the SRC](#)
    - \*6.1.1. [Handling media stream updates](#)
  - \*6.2. [Procedures at the SRS](#)
- \*7. [RTP Handling](#)
- \*8. [Metadata](#)
  - \*8.1. [Procedures at the SRC](#)
  - \*8.2. [Procedures at the SRS](#)
    - \*8.2.1. [Formal Syntax](#)
- \*9. [Persistent Recording](#)
- \*10. [Extensions for Recording-aware User Agents](#)
  - \*10.1. [Procedures at the record-aware user agent](#)
    - \*10.1.1. [Recording preference](#)
  - \*10.2. [Procedures at the SRC](#)

- \*10.2.1. [Recording indication](#)
- \*10.2.2. [Recording preference](#)
- \*11. [IANA Considerations](#)
- \*11.1. [Registration of Option Tags](#)
- \*11.1.1. [siprec Option Tag](#)
- \*11.1.2. [record-aware Option Tag](#)
- \*11.2. [Registration of media feature tags](#)
- \*11.2.1. [src feature tag](#)
- \*11.2.2. [srs feature tag](#)
- \*11.3. [New Content-Disposition Parameter Registrations](#)
- \*11.4. [Media Type Registration](#)
- \*11.4.1. [Registration of MIME Type application/rs-metadata](#)
- \*11.4.2. [Registration of MIME Type application/rs-metadata-request](#)
- \*11.5. [SDP Attributes](#)
- \*11.5.1. ['record' SDP Attribute](#)
- \*11.5.2. ['recordpref' SDP Attribute](#)
- \*12. [Security Considerations](#)
- \*12.1. [Authentication and Authorization](#)
- \*13. [Acknowledgements](#)
- \*14. [References](#)
- \*14.1. [Normative References](#)
- \*14.2. [Informative References](#)
- \*[Authors' Addresses](#)

## **[1. Introduction](#)**

This document specifies the mechanism to record a Communication Session (CS) by delivering real-time media and metadata from the CS to a recording device. In accordance to the architecture [\[I-D.ietf-siprec-](#)

[architecture](#)], the Session Recording Protocol specifies the use of SIP, SDP, and RTP to establish a Recording Session (RS) from the Session Recording Client (SRC), which is on the path of the CS, to a Session Recording Server (SRS) at the recording device.

SIP is also used to deliver metadata to the recording device, as specified in [\[I-D.ietf-siprec-metadata\]](#). Metadata is information that describes recorded media and the CS to which they relate.

The Session Recording Protocol intends to satisfy the SIP-based Media Recording requirements listed in [\[RFC6341\]](#).

## **[2. Definitions](#)**

This document refers to the core definitions provided in the architecture document [\[I-D.ietf-siprec-architecture\]](#).

## **[3. Scope](#)**

The scope of the Session Recording Protocol includes the establishment of the recording sessions and the reporting of the metadata. The scope also includes extensions supported by User Agents participating in the CS such as indication of recording. The user agents need not be recording-aware in order to participate in a CS being recorded. The following items, which are not an exhaustive list, do not represent the protocol itself and are considered out of the scope of the Session Recording Protocol:

- \*Delivering recorded media in real-time as the CS media
- \*Specifications of criteria to select a specific CS to be recorded or triggers to record a certain CS in the future
- \*Recording policies that determine whether the CS should be recorded and whether parts of the CS are to be recorded
- \*Retention policies that determine how long a recording is stored
- \*Searching and accessing the recorded media and metadata
- \*Policies governing how CS users are made aware of recording
- \*Delivering additional recording session metadata through non-SIP mechanism

## **[4. Overview of operations](#)**

This section is informative and provides a description of recording operations.

As mentioned in the architecture document [\[I-D.ietf-siprec-architecture\]](#), there are a number of types of call flows based on the location of the Session Recording Client. The following sample call

flows provide a quick overview of the operations between the SRC and the SRS.

#### **4.1. Delivering recorded media**

When a SIP Back-to-back User Agent (B2BUA) with SRC functionality routes a call from UA(A) to UA(B), the SRC has access to the media path between the user agents. When the SRC is aware that it should be recording the conversation, the SRC can cause the B2BUA to bridge the media between UA(A) and UA(B). The SRC then establishes the Recording Session with the SRS and sends replicated media towards the SRS.

An endpoint may also have SRC functionality, where the endpoint itself establishes the Recording Session to the SRS. Since the endpoint has access to the media in the Communication Session, the endpoint can send replicated media towards the SRS.

The following is a sample call flow that shows the SRC establishing a recording session towards the SRS. The call flow is essentially identical when the SRC is a B2BUA or as the endpoint itself. Note that the SRC can choose when to establish the Recording Session independent of the Communication Session, even though the following call flow suggests that the SRC is establishing the Recording Session (message #5) after the Communication Session is established.

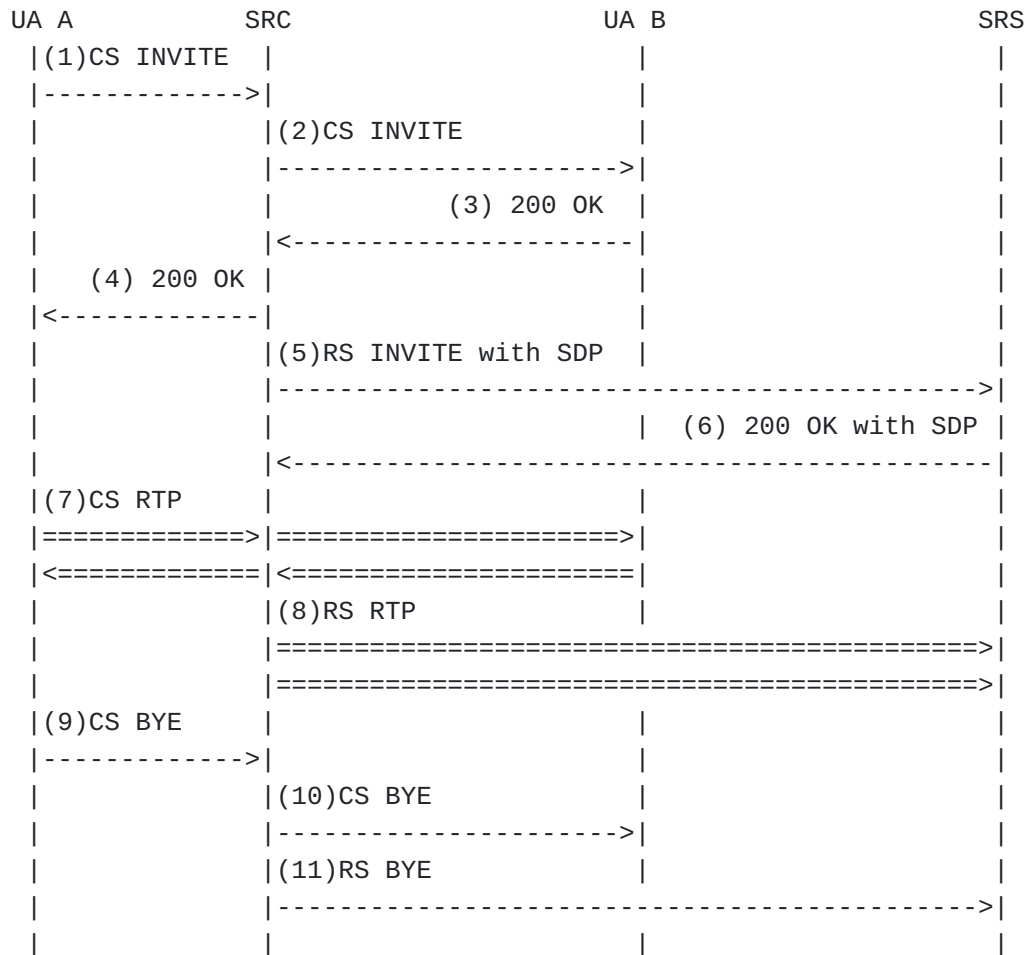


Figure 1: Basic Recording Call flow

The above call flow can also apply to the case of a centralized conference with a mixer. For clarity, ACKs to INVITES and 200 OKs to BYEs are not shown. The conference focus can provide the SRC functionality since the conference focus has access to all the media from each conference participant. When a recording is requested, the SRC delivers the metadata and the media streams to the SRS. Since the conference focus has access to a mixer, the SRC may choose to mix the media streams from all participants as a single mixed media stream towards the SRS.

An SRC can use a single recording session to record multiple communication sessions. Every time the SRC wants record a new call, the SRC updates the recording session with a new SDP offer to add new recorded streams to the recording session, and correspondingly also update the metadata for the new call.

## **4.2. Delivering recording metadata**

The SRC is responsible for the delivery of metadata to the SRS. The SRC may provide an initial metadata snapshot about recorded media streams in the initial INVITE content in the recording session. Subsequent metadata updates can be represented as a stream of events in UPDATE or reINVITE requests sent by the SRC. These metadata updates are normally incremental updates to the initial metadata snapshot to optimize on the size of updates, however, the SRC may also decide to send a new metadata snapshot anytime.

Metadata is transported in the body of INVITE or UPDATE messages. Certain metadata, such as the attributes of the recorded media stream are located in the SDP of the recording session.

The SRS has the ability to send a request to the SRC to request for a new metadata snapshot update from the SRC. This can happen when the SRS fails to understand the current stream of incremental updates for whatever reason, for example, when SRS loses the current state due to internal failure. The SRS may optionally attach a reason along with the snapshot request. This request allows both SRC and SRS to restart the states with a new metadata snapshot so that further metadata incremental updates will be based on the latest metadata snapshot. Similar to the metadata content, the metadata snapshot request is transported as content in UPDATE or INVITE sent by the SRS in the recording session.

SRC	SRS
(1) INVITE (metadata snapshot)	
----->	
	(2) 200 OK
<-----	
(3) ACK	
----->	
(4) RTP	
=====>	
=====>	
(5) UPDATE (metadata update 1)	
----->	
	(6) 200 OK
<-----	
(7) UPDATE (metadata update 2)	
----->	
	(8) 200 OK
<-----	
(9) UPDATE (metadata snapshot request)	
<-----	
	(10) 200 OK
----->	
(11) INVITE (metadata snapshot 2 + SDP offer)	
----->	
	(12) 200 OK (SDP answer)
<-----	
(13) UPDATE (metadata update 1 based on snapshot 2)	
----->	
	(14) 200 OK
<-----	

Figure 3: Delivering metadata via SIP UPDATE

## 5. Initiating a Recording Session

### 5.1. Procedures at the SRC

The SRC can initiate a recording session by sending a SIP INVITE request to the SRS. In this case, the From header MUST contain the identity of the SRC, and the To header MUST contain the identity of the SRS. Participants information is not recorded in the From or To header; they are included in the metadata.

The SRC MUST include the 'src' feature tag in the Contact URI, defined in this specification as an extension to [\[RFC3840\]](#), for all recording sessions. An SRS uses the presence of the 'src' feature tag in dialog



creating and modifying requests and responses to confirm that the dialog being created is for the purpose of a Recording Session. In addition, a registrar could discover that a UA is an SRC based on the presence of this feature tag in a registration. Other SIP Recording extensions and behaviors can be triggered by the presence of this feature tag.

Since SIP Caller Preferences extensions are optional to implement for routing proxies, there is no guarantee that a recording session will be routed to an SRC or SRS. A new options tag is introduced: "siprec". As per [\[RFC3261\]](#), only an SRC or an SRS can accept this option tag in a recording session. An SRC SHOULD include the "siprec" option tag in the Require header when initiating a Recording Session so that other types of user agents can simply reject the INVITE request with a 420 Bad Extension.

## **5.2. Procedures at the SRS**

The SRS can initiate a recording session by sending a SIP INVITE request to the SRC. In this case, the From header MUST contain the identity of the SRS, and the To header MUST contain the identity of the SRC.

The SRS MUST include the 'srs' feature tag in the Contact URI, as per [\[RFC3840\]](#), for all recording sessions. An SRC uses the presence of this feature tag in dialog creating and modifying requests and responses to confirm that the dialog being created is for the purpose of a Recording Session (REQ-30). In addition, a registrar could discover that a UA is an SRS based on the presence of this feature tag in a registration. Other SIP Recording extensions and behaviors can be triggered by the presence of this feature tag.

An SRS SHOULD include the "siprec" option tag in the Require header as per [\[RFC3261\]](#) when initiating a Recording Session so that other types of user agents can simply reject the INVITE request with a 420 Bad Extension.

## **6. SDP Handling**

The SRC and SRS follows the SDP offer/answer model in [\[RFC3264\]](#). The rest of this section describes conventions used in a recording session.

### **6.1. Procedures at the SRC**

Since the SRC does not expect to receive media from the SRS, the SRC typically sets each media stream of the SDP offer to only send media, by qualifying them with the a=sendonly attribute, according to the procedures in [\[RFC3264\]](#).

The SRC sends recorded streams of participants to the SRS, and the SRC MUST provide a label attribute (a=label), as per [\[RFC4574\]](#), on each media stream in order to identify the recorded stream with the rest of the metadata. The a=label attribute identifies each recorded media

stream, and the label name is mapped to the Media Stream Reference in the metadata in [\[I-D.ietf-siprec-metadata\]](#). Note that a recorded stream is different than a CS stream; the metadata provides a list of participants that contributes to each recorded stream. The following is an example of SDP with both audio and video recorded streams. Note that the following example contain unfolded lines longer than 72 characters. These are captured between <allOneLine> tags.

```
v=0
o=SRS 2890844526 2890844526 IN IP4 198.51.100.1
s=-
c=IN IP4 198.51.100.1
t=0 0
m=audio 12240 RTP/AVP 0 4 8
a=sendonly
a=label:1
m=video 22456 RTP/AVP 98
a=rtpmap:98 H264/90000
<allOneLine>
a=fmtp:98 profile-level-id=42A01E;
        sprop-parameter-sets=Z0IACpZTBmI,aMljiA==
</allOneLine>
a=sendonly
a=label:2
m=audio 12242 RTP/AVP 0 4 8
a=sendonly
a=label:3
m=audio 22458 RTP/AVP 98
a=rtpmap:98 H264/90000
<allOneLine>
a=fmtp:98 profile-level-id=42A01E;
        sprop-parameter-sets=Z0IACpZTBmI,aMljiA==
</allOneLine>
a=sendonly
a=label:4
```

Figure 4: Sample SDP with audio and video streams

#### **6.1.1. Handling media stream updates**

Over the lifetime of a recording session, the SRC can add and remove recorded streams from the recording session for various reasons. For example, when a CS stream is added or removed from the CS, or when a CS is created or terminated if a recording session handles multiple CSes. To remove a recorded stream from the recording session, the SRC sends a new SDP offer where the port of the media stream to be removed is set

to zero, according to the procedures in [\[RFC3264\]](#). To add a recorded stream to the recording session, the SRC sends a new SDP offer by adding a new media stream description or by reusing an old media stream which had been previously disabled, according to the procedures in [\[RFC3264\]](#).

The SRC can temporarily discontinue streaming and collection of recorded media from the SRC to the SRS for reason such as masking the recording. In this case, the SRC sends a new SDP offer and sets the media stream to inactive (a=inactive) for each recorded stream to be paused, as per the procedures in [\[RFC3264\]](#). To resume streaming and collection of recorded media, the SRC sends a new SDP offer and sets the media streams with a=sendonly attribute. Note that when a CS stream is muted/unmuted, this information is conveyed in the metadata by the SRC. The SRC SHOULD not modify the media stream with a=inactive for mute since this operation is reserved for pausing the RS media.

## **6.2. Procedures at the SRS**

The SRS only receives RTP streams from the SRC, the SDP answer normally sets each media stream to receive media, by setting them with the a=recvonly attribute, according to the procedures of [\[RFC3264\]](#). When the SRS is not ready to receive a recorded stream, the SRS sets the media stream as inactive in the SDP offer or answer by setting it with a=inactive attribute, according to the procedures of [\[RFC3264\]](#). When the SRS is ready to receive recorded streams, the SRS sends a new SDP offer and sets the media streams with a=recvonly attribute.

The following sequence diagram shows an example where the SRS is initially not ready to receive recorded streams, and later updates the recording session when the SRS is ready to record.

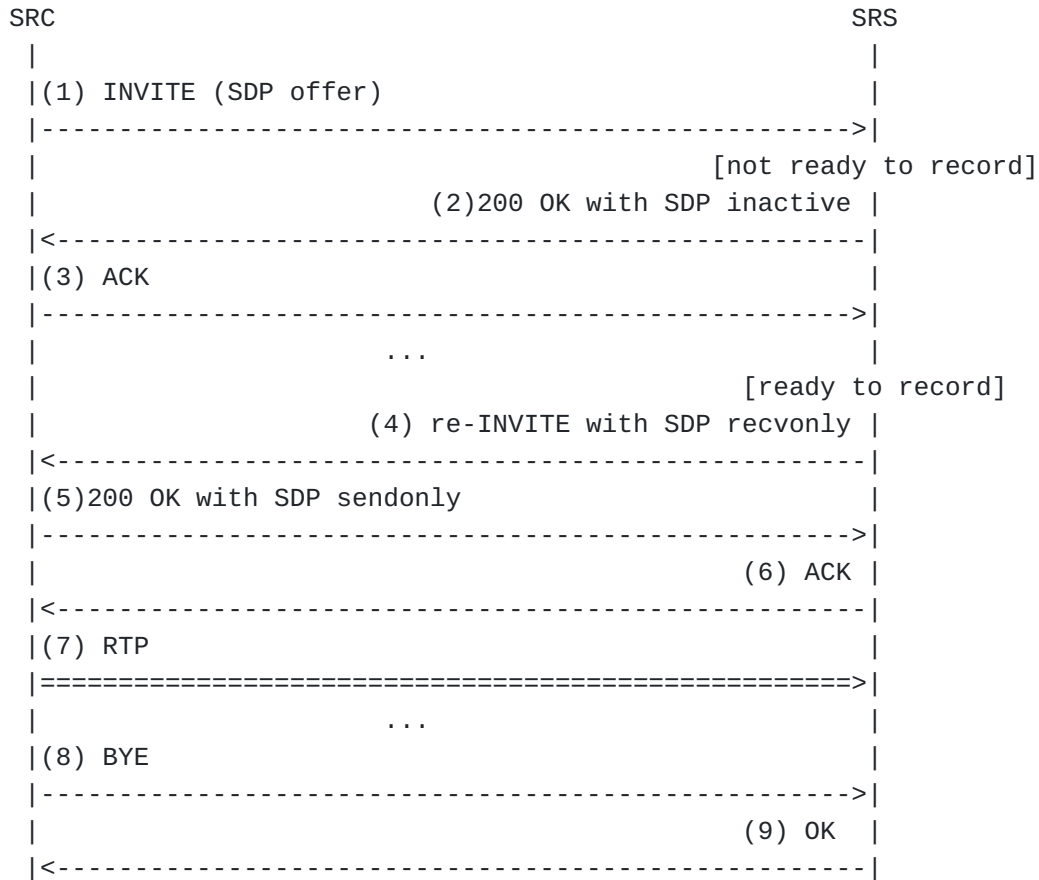


Figure 5: SRS to offer with a=inactive

## 7. RTP Handling

This is a placeholder section to specify any protocol impacts or recommendations for RTP usage in the session recording protocol. The details are listed in [\[I-D.eckel-siprec-rtp-rec\]](#)

## 8. Metadata

### 8.1. Procedures at the SRC

The SRC is responsible to deliver metadata to the SRS in a recording session. Metadata can be provided by the SRC in the initial INVITE request when establishing the recording session, and subsequent metadata updates can be provided by the SRC in reINVITE and UPDATE requests and responses in the recording session.

Certain metadata attributes are contained in the SDP, and others are contained in a new content type "application/rs-metadata". The format of the metadata is described as part of the mechanism in [\[I-D.ietf-siprec-metadata\]](#). A new "disposition-type" of Content-Disposition is defined for the purpose of carrying metadata and the value is

"recording-session". The "recording-session" value indicates that the "application/rs-metadata" content contains metadata to be handled by the SRS, and the disposition can be carried in either INVITE or UPDATE requests or responses sent by the SRC.

Metadata sent by the SRC can be categorized as either a full metadata snapshot or partial update. A full metadata snapshot describes all the recorded streams and all metadata associated with the recording session. When the SRC sends a full metadata snapshot, the SRC MUST send an INVITE request with an SDP offer and the "recording-session" disposition. A partial update represents an incremental update since the last metadata update sent by the SRC. A partial update sent by the SRC can be an INVITE request or response with an SDP offer, or an INVITE/UPDATE request or response containing a "recording-session" disposition, or an INVITE request containing both an SDP offer and the "recording-session" disposition.

The following is an example of a full metadata snapshot sent by the SRC in the initial INVITE request:

```

INVITE sip:recorder@example.com SIP/2.0
Via: SIP/2.0/TCP src.example.com;branch=z9hG4bKdf6b622b648d9
From: <sip:2000@example.com>;tag=35e195d2-947d-4585-946f-098392474
To: <sip:recorder@example.com>
Call-ID: d253c800-b0d1ea39-4a7dd-3f0e20a
CSeq: 101 INVITE
Max-Forwards: 70
Require: siprec
Accept: application/sdp, application/rs-metadata,
       application/rs-metadata-request
Contact: <sip:2000@src.example.com>;src
Content-Type: multipart/mixed;boundary=foobar
Content-Length: [length]

--foobar
Content-Type: application/sdp

v=0
o=SRS 2890844526 2890844526 IN IP4 198.51.100.1
s=-
c=IN IP4 198.51.100.1
t=0 0
m=audio 12240 RTP/AVP 0 4 8
a=sendonly
a=label:1

--foobar
Content-Type: application/rs-metadata
Content-Disposition: recording-session

[metadata content]

```

Figure 6: Sample INVITE request for the recording session

## **8.2. Procedures at the SRS**

The SRS receives metadata updates from the SRC in INVITE and UPDATE requests. Since the SRC can send partial updates based on the previous update, the SRS needs to keep track of the sequence of updates from the SRC.

In the case of an internal failure at the SRS, the SRS may fail to recognize a partial update from the SRC. The SRS may be able to recover from the internal failure by requesting for a full metadata snapshot from the SRC. Certain errors, such as syntax errors or semantic errors in the metadata information, are likely caused by an error on the SRC side, and it is likely the same error will occur again even when a full metadata snapshot is requested. In order to avoid repeating the same

error, it is RECOMMENDED that the SRS terminate the recording session when a syntax error or semantic error is detected in the metadata. When the SRS requires a full metadata snapshot, the SRS sends a metadata snapshot request to the SRC in an INVITE/UPDATE request or in an INVITE/UPDATE response. The metadata snapshot request is contained the content with the disposition type "recording-session". The format of the content is "application/rs-metadata-request", and the body format is chosen to be a simple text-based format. The following shows an example:

```
UPDATE sip:2000@src.exmaple.com SIP/2.0
Via: SIP/2.0/UDP srs.example.com;branch=z9hG4bKdf6b622b648d9
To: <sip:2000@exmaple.com>;tag=35e195d2-947d-4585-946f-098392474
From: <sip:recorder@example.com>;tag=1234567890
Call-ID: d253c800-b0d1ea39-4a7dd-3f0e20a
CSeq: 1 UPDATE
Max-Forwards: 70
Require: siprec
Contact: <sip:recorder@srs.example.com>;srs
Accept: appliation/sdp, application/rs-metadata
Content-Disposition: recording-session
Content-Type: application/rs-metadata-request
Content-Length: [length]
```

SRS internal error

Figure 7: Metadata Request

The SRS MAY include the reason why a metadata snapshot request is being made to the SRC in the reason line. This reason line is free form text, mainly designed for logging purposes on the SRC side. The processing of the content by the SRC is entirely optional since the content is for logging only, and the snapshot request itself is indicated by the use of the application/rs-metadata-request content type. When the SRC receives the request for a metadata snapshot, the SRC MUST provide a full metadata snapshot in a separate INVITE transaction, along with an SDP offer. All subsequent metadata updates sent by the SRC MUST be based on the new metadata snapshot.

#### **8.2.1. Formal Syntax**

The formal syntax for the application/rs-metadata-request MIME is described below using the augmented Backus-Naur Form (BNF) as described in [\[RFC2234\]](#).

```
snapshot-request = srs-reason-line CRLF
srs-reason-line = [TEXT-UTF8-TRIM]
```

## **9. Persistent Recording**

Persistent recording is a specific use case outlined in REQ-005 or Use Case 4 in [\[RFC6341\]](#), where a recording session can be established in the absence of a communication session. The SRC continuously sends recorded media to the SRS in the absence of a CS for certain allocated devices; allocated devices can include a specific physical device, a specific person or contact registered, or a set of trunks or ports of a gateway. To continuously record, the SRC adds recorded streams into the recording session for all devices with persistent recording. By allocating recorded streams and continuously sending recorded media to the SRS, the SRC does not have to prepare new recorded streams with new SDP offer when a new communication session is created and also does not impact the timing of the CS. The SRC only needs to update the metadata when new communication sessions are created.

When there is no communication sessions running on the devices with persistent recording, there is no recorded media to stream from the SRC to the SRS. In certain environments where Network Address Translator (NAT) is used, typically a minimum of flow activity is required to maintain the NAT binding for each port opened. In order not to lose the NAT bindings for the RTP/RTCP ports opened for the recorded streams, the SRC and SRS SHOULD follow the recommendations provided in [\[RFC6263\]](#) to maintain the NAT bindings.

## **10. Extensions for Recording-aware User Agents**

The following sections describe the SIP and SDP extensions for recording-aware user agents.

### **10.1. Procedures at the record-aware user agent**

A recording-aware UA SHOULD indicate that it can accept reporting of recording indication provided by the SRC. A new option tag "record-aware" is introduced to indicate such awareness. The recording-aware UA SHOULD include the "record-aware" option tag in the Supported header when initiating or establishing a CS. A recording-aware UA that has indicated recording awareness MUST provide at recording indication to the end user through an appropriate user interface an indication whether recording is on or off for a given medium based on the most recently received a=record SDP attribute for that medium.

Some user agents that are automatons (eg. IVR, media server, PSTN gateway) may not have a user interface to render recording indication. When such user agent indicates recording awareness, the UA SHOULD render recording indication through other means, such as passing an inband tone on the PSTN gateway, putting the recording indication in a log file, or raising an application event in a VoiceXML dialog. These user agents MAY also choose not to indicate recording awareness, thereby relying on whatever mechanism an SRC chooses to indicate recording, such as playing a tone inband.



#### 10.1.1. Recording preference

A recording-aware UA involved in a CS MAY request the CS to be recorded or not recorded. This indication of recording preference can be sent at session establishment time or during the session.

A new SDP attribute "recordpref" is introduced. The SDP attribute appears at the media level and can appear in an SDP offer or answer. The recording indication applies to the specified media stream only. The following is the ABNF of the recordpref attribute:

```
recordpref-attr = "a=recordpref:" pref
```

```
pref = "on" / "off" / "pause" / "nopreference"
```

**on** Request for recording if it has not already been started. If the recording is currently paused, request to resume recording.

**off** Request for no recording. If recording has already been started, then this preference indicates a request to stop recording.

**pause** Request to pause recording if recording is currently in progress.

**nopreference** To indicate that the UA has no preference on recording. While the absence of this attribute indirectly implies the lack of preference, using this value allows the UA to explicitly state no preference to being recorded.

#### 10.2. Procedures at the SRC

When a UA has indicated that it is recording-aware through the "record-aware" option tag, the SRC MUST provide recording indications in a new SDP attribute described in the following section. In the absence of the "record-aware" option tag, meaning that the UA is not recording-aware, an SRC MUST provide recording indications, where SRC is required to do so based on policies, through other means such as playing a tone inband.

##### 10.2.1. Recording indication

While there are existing mechanisms for providing an indication that a CS is being recorded, these mechanisms are usually delivered on the CS media streams such as playing an in-band tone or an announcement to the participants. A new SDP attribute is introduced to allow a recording-aware UA to render recording indication at the user interface.

The 'record' SDP attribute appears at the media level in either SDP offer or answer. The recording indication applies to the specified media stream only, for example, only the audio portion of the call is recorded in an audio/video call. The following is the ABNF of the 'record' attribute:

attribute /= record-attr

; attribute defined in RFC 4566

record-attr = "record:" indication

indication = "on" / "off" / "paused"

**on** Recording is in progress.

**off** No recording is in progress.

**paused** Recording is in progress by media is paused.

The recording attribute is a declaration by an endpoint in the CS to indicate whether recording is taking place. For example, if a UA (A) is initiating a call to UA (B) and UA (A) is also an SRC that is performing the recording, then UA (A) provides the recording indication in the SDP offer with a=record:on. When UA (B) receives the SDP offer, UA (B) will see that recording is happening on the other endpoint of this session. If UA (B) does not wish to perform recording itself, UA (B) provides the recording indication as a=record:off in the SDP answer.

Whenever the recording indication needs to change, such as termination of recording, then the UA MUST initiate a reINVITE to update the SDP attribute to a=record:off. The following call flow shows an example of the offer/answer with the recording indication attribute.

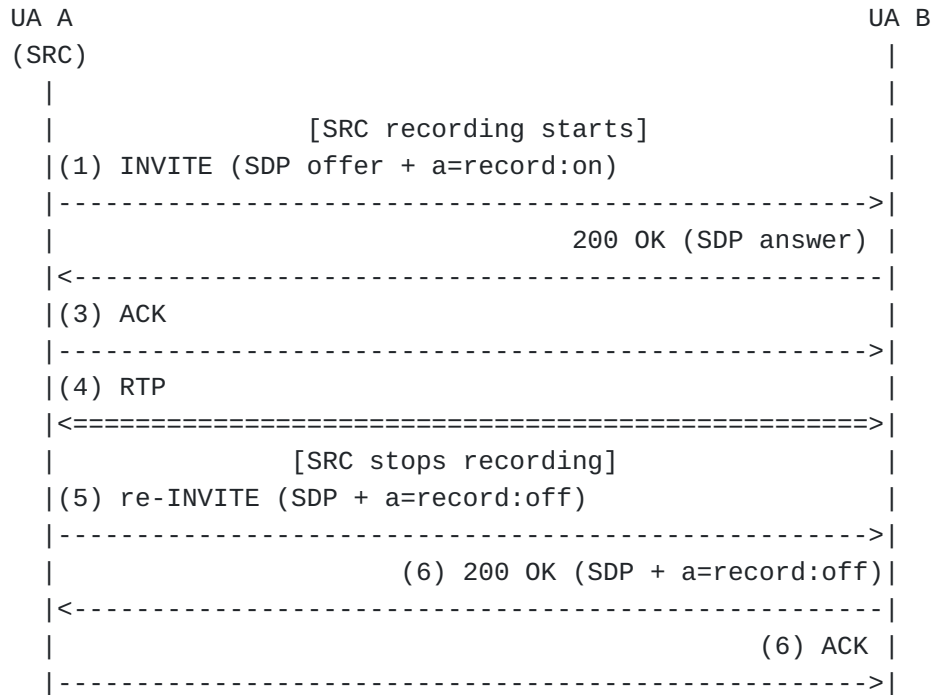


Figure 8: Recording indication example

If a call is traversed through one or more SIP B2BUA, and it happens that there are more than one SRC in the call path, the recording indication attribute does not provide any hint as to which SRC is performing the recording, meaning the endpoint only knows that the call is being recorded. This attribute is also not used as an indication to negotiate which SRC in the call path will perform recording and is not used as a request to start/stop recording if there are multiple SRCs in the call path.

### [10.2.2. Recording preference](#)

When the SRC receives the `a=recordpref` SDP in an SDP offer or answer, the SRC MAY choose to honor such request to record the request based on local policy on the SRC. When the SRC honors the request, the SRC MUST also update the recording indication to reflect the current state of the recording (on/off/paused).

## [11. IANA Considerations](#)

### [11.1. Registration of Option Tags](#)

This specification registers two option tags. The required information for this registration, as specified in [\[RFC3261\]](#), is as follows.

#### **11.1.1. siprec Option Tag**

\*Name: siprec

\*Description: This option tag is for identifying the SIP session for the purpose of recording session only. This is typically not used in a Supported header. When present in a Require header in a request, it indicates that the UAS MUST be either a SRC or SRS capable of handling the contexts of a recording session.

#### **11.1.2. record-aware Option Tag**

\*Name: record-aware

\*Description: This option tag is to indicate the ability for the user agent to receive recording indicators in media level SDP. When present in a Supported header, it indicates that the UA can receive recording indicators in media level SDP.

### **11.2. Registration of media feature tags**

This document registers two new media feature tags in the SIP tree per the process defined in [\[RFC2506\]](#) and [\[RFC3840\]](#)

#### **11.2.1. src feature tag**

\*Media feature tag name: sip.src

\*ASN.1 Identifier: 25

\*Summary of the media feature indicated by this tag: This feature tag indicates that the user agent is a Session Recording Client for the purpose for Recording Session.

\*Values appropriate for use with this feature tag: boolean

\*The feature tag is intended primarily for use in the following applications, protocols, services, or negotiation mechanisms: This feature tag is only useful for a Recording Session.

\*Examples of typical use: Routing the request to a Session Recording Server.

\*Security Considerations: Security considerations for this media feature tag are discussed in Section 11.1 of RFC 3840.

#### **11.2.2. srs feature tag**

\*Media feature tag name: sip.srs

\*ASN.1 Identifier: 26

\*Summary of the media feature indicated by this tag: This feature tag indicates that the user agent is a Session Recording Server for the purpose for Recording Session.

\*Values appropriate for use with this feature tag: boolean

\*The feature tag is intended primarily for use in the following applications, protocols, services, or negotiation mechanisms: This feature tag is only useful for a Recording Session.

\*Examples of typical use: Routing the request to a Session Recording Client.

\*Security Considerations: Security considerations for this media feature tag are discussed in Section 11.1 of RFC 3840.

### **11.3. New Content-Disposition Parameter Registrations**

This document registers a new "disposition-type" value in Content-Disposition header: recording-session.  
recording-session the body describes the metadata information about the recording session

### **11.4. Media Type Registration**

#### **11.4.1. Registration of MIME Type application/rs-metadata**

This document registers the application/rs-metadata MIME media type in order to describe the recording session metadata. This media type is defined by the following information:  
Media type name: application  
Media subtype name: rs-metadata  
Required parameters: none  
Options parameters: none

#### **11.4.2. Registration of MIME Type application/rs-metadata-request**

This document registers the application/rs-metadata-request MIME media type in order to describe a recording session metadata snapshot request. This media type is defined by the following information:  
Media type name: application  
Media subtype name: rs-metadata-request  
Required parameters: none  
Options parameters: none

### **11.5. SDP Attributes**

This document registers the following new SDP attributes.

#### **11.5.1. 'record' SDP Attribute**

Contact names: Leon Portman [leon.portman@nice.com](mailto:leon.portman@nice.com), Henry Lum  
[henry.lum@genesyslab.com](mailto:henry.lum@genesyslab.com)

Attribute name: record

Long form attribute name: Recording Indication

Type of attribute: media level

Subject to charset: no

This attribute provides the recording indication for the session or media stream.

Allowed attribute values: on, off, paused

#### **11.5.2. 'recordpref' SDP Attribute**

Contact names: Leon Portman [leon.portman@nice.com](mailto:leon.portman@nice.com), Henry Lum  
[henry.lum@genesyslab.com](mailto:henry.lum@genesyslab.com)

Attribute name: recordpref

Long form attribute name: Recording Preference

Type of attribute: media level

Subject to charset: no

This attribute provides the recording indication for the session or media stream.

Allowed attribute values: on, off, pause, nopreference

### **12. Security Considerations**

The recording session is fundamentally a standard SIP dialog [\[RFC3261\]](#), therefore, the recording session can reuse any of the existing SIP security mechanism available for securing the recorded media as well as metadata. Other security considerations are outlined in the use cases and requirements document [\[RFC6341\]](#).

#### **12.1. Authentication and Authorization**

The recording session reuses the SIP mechanism to challenge requests that is based on HTTP authentication. The mechanism relies on 401 and 407 SIP responses as well as other SIP header fields for carrying challenges and credentials.

The SRS may have its own set of recording policies to authorize recording requests from the SRC. The use of recording policies is outside the scope of the Session Recording Protocol.

### **13. Acknowledgements**

We want to thank John Elwell, Paul Kyzivat, Partharsarathi R, Ram Mohan R, Charles Eckel, Hadriel Kaplan, Adam Roach, Miguel Garcia for their valuable comments and inputs to this document.

## 14. References

### 14.1. Normative References

[RFC2506]	<a href="#">Holtman, K., Mutz, A. and T. Hardie, "Media Feature Tag Registration Procedure"</a> , BCP 31, RFC 2506, March 1999.
[RFC2119]	<a href="#">Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels"</a> , BCP 14, RFC 2119, March 1997.
[RFC2234]	<a href="#">Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF"</a> , RFC 2234, November 1997.
[RFC2804]	IABIESG, <a href="#">"IETF Policy on Wiretapping"</a> , RFC 2804, May 2000.
[RFC3261]	Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M. and E. Schooler, <a href="#">"SIP: Session Initiation Protocol"</a> , RFC 3261, June 2002.
[RFC3264]	Rosenberg, J. and H. Schulzrinne, <a href="#">"An Offer/Answer Model with Session Description Protocol (SDP)"</a> , RFC 3264, June 2002.
[RFC4574]	Levin, O. and G. Camarillo, <a href="#">"The Session Description Protocol (SDP) Label Attribute"</a> , RFC 4574, August 2006.
[RFC3840]	Rosenberg, J., Schulzrinne, H. and P. Kyzivat, <a href="#">"Indicating User Agent Capabilities in the Session Initiation Protocol (SIP)"</a> , RFC 3840, August 2004.
[RFC3841]	Rosenberg, J., Schulzrinne, H. and P. Kyzivat, <a href="#">"Caller Preferences for the Session Initiation Protocol (SIP)"</a> , RFC 3841, August 2004.
[RFC6341]	Rehor, K., Portman, L., Hutton, A. and R. Jain, <a href="#">"Use Cases and Requirements for SIP-Based Media Recording (SIPREC)"</a> , RFC 6341, August 2011.
[I-D.ietf-siprec-metadata]	R, R, Ravindran, P and P Kyzivat, <a href="#">"Session Initiation Protocol (SIP) Recording Metadata"</a> , Internet-Draft draft-ietf-siprec-metadata-05, October 2011.

### 14.2. Informative References

[I-D.ietf-siprec-architecture]	Hutton, A, Portman, L, Jain, R and K Rehor, <a href="#">"An Architecture for Media Recording using the Session Initiation Protocol"</a> , Internet-Draft draft-ietf-siprec-architecture-03, October 2011.
[I-D.eckel-siprec-rtp-rec]	Eckel, C, <a href="#">"Real-time Transport Protocol (RTP) Recommendations for SIPREC"</a> , Internet-Draft draft-eckel-siprec-rtp-rec-03, October 2011.
[RFC4508]	

	Levin, O. and A. Johnston, " <a href="#">Conveying Feature Tags with the Session Initiation Protocol (SIP) REFER Method</a> ", RFC 4508, May 2006.
[RFC4579]	Johnston, A. and O. Levin, " <a href="#">Session Initiation Protocol (SIP) Call Control - Conferencing for User Agents</a> ", BCP 119, RFC 4579, August 2006.
[RFC6263]	Marjou, X. and A. Sollaud, " <a href="#">Application Mechanism for Keeping Alive the NAT Mappings Associated with RTP / RTP Control Protocol (RTCP) Flows</a> ", RFC 6263, June 2011.

### Authors' Addresses

Leon Portman editor Portman NICE Systems 8 Hapnina Ra'anana, 43017 Israel EMail: [leon.portman@nice.com](mailto:leon.portman@nice.com)

Henry Lum editor Lum Genesys, Alcatel-Lucent 1380 Rodick Road, Suite 200 Markham, Ontario L3R4G5 Canada EMail: [henry.lum@genesyslab.com](mailto:henry.lum@genesyslab.com)

Alan Johnston Johnston Avaya St. Louis, MO 63124 EMail: [alan.b.johnston@gmail.com](mailto:alan.b.johnston@gmail.com)

Andrew Hutton Hutton Siemens Enterprise Communications EMail: [andrew.hutton@siemens-enterprise.com](mailto:andrew.hutton@siemens-enterprise.com)