Abstract

RFC 2326 [1] overloads the functionality of the PLAY and PAUSE methods to provide rudimentary support for "muting" individual streams in an (Real Time Streaming Protocol) presentation. The current specification has several inconsistencies. In addition, we believe that this functionality should be optional. This document specifies an extension to RTSP consisting of new MUTE and UNMUTE methods, providing equivalent functionality in a consistent, optional manner.
1. Introduction

MUTE and UNMUTE are new optional RTSP methods designed to replace the functionality specified in [1] of doing PAUSE <stream-url> during an aggregate RTSP presentation. Clients MAY implement these methods to temporarily suppress delivery of one or more streams in an RTSP presentation without affecting delivery of the other streams.
2. Method definitions and examples

2.1 MUTE

The MUTE method causes the server to temporarily suspend delivery of the requested stream. The URL must be for a stream which has already been SETUP in the current session. If aggregate control is in use, the client MUST NOT send the presentation URL. The presentation clock and any streams which are still playing are not affected.

MUTE is legal in the Ready and Playing states. If the stream was muted and is subsequently PAUSEd or PLAYed, it remains muted. Subsequent PLAY and PAUSE requests MUST NOT implicitly mute or unmute the stream.

The Range header may be used to schedule MUTE operations to occur at a particular time. If so used, the header MUST NOT contain a closed range. If the starting time is earlier than the current time, or no Range header was sent, the MUTE acts immediately. If the starting time is later than the current time, the MUTE does not act until that time. If the Range header specifies a time outside the current PLAY request, or a Range header is specified when sending a MUTE in the Ready state, the error "457 Invalid Range" is returned.

If a MUTE request is scheduled for future execution, but a subsequent PAUSE request halts stream delivery before the scheduled MUTE point is reached, the server MUST NOT execute the pending MUTE request.

A MUTE request should override any previous MUTE request for the same stream that are currently active or pending.

If the stream is being delivered over RTP [2], RTCP reports SHOULD continue to be exchanged while the stream is muted. This allows for synchronization to be maintained once the stream is unmuted so that loosely coupled RTP receivers will behave normally once the stream is unmuted.

2.2 UNMUTE

The UNMUTE request causes the server to resume delivery of a stream that has previously been muted using the MUTE request. If aggregate control is in use, the client MUST NOT send the presentation URL. Each stream must be unmuted individually.

If a server implements MUTE, it MUST also implement UNMUTE.

If the session is in the Playing state, the stream is resumed at the point where it would be had the MUTE request not been sent. If
aggregate control is in use, the stream MUST maintain the same
synchronization it previously had with the rest of the streams in the
presentation. If the presentation is in the Ready state, the stream
is taken out of the muted state and any future PLAY requests will
deliver the stream normally.

The Range header may be used to schedule UNMUTE operations to occur
at a particular time. If so used, the header MUST NOT contain a
closed range. If the starting time is earlier than the current time,
or no Range header was sent, the UNMUTE acts immediately. If the
starting time is later than the current time, the UNMUTE does not act
until that time. If the Range header specifies a time outside the
current PLAY request, or a Range header is specified when sending a
UNMUTE in the Ready state, the error "457 Invalid Range" is returned.

If the stream is being delivered over RTP [2], the same RTP session
is used after the stream as unmuted as was used before the stream was
muted. Sequence numbers MUST continue in sequence (the server does
not skip sequence numbers for the packets not delivered due to the
MUTE). RTP timestamps must continue with the presentation time (the
timestamps for the packets delivered after UNMUTE are the same as
they would have been had the stream not been muted). The server
SHOULD send an RTP-Info header with the UNMUTE response to inform the
client where the unmuted packets begin.
3. Negotiation

Several mechanisms are available for clients and servers to negotiate their implementation of this extension.

The client may send an OPTIONS request for one of the stream URLs. The server’s Allow header will include MUTE and UNMUTE if the server supports them.

The client may attempt a MUTE request. If the server implements RTSP properly but does not implement the MUTE extension, it will respond with "501 Not Implemented" but not terminate the session or the connection.

The client may use the option tag "mute" in the Require header on a SETUP request. This may be useful if the client intends to use MUTE to switch between one or more streams with different representations of the same data. For example, if the client intends to use MUTE to switch between multiple language tracks, it can send "Require: mute" on all SETUP requests except for the user’s current default language. If the server does not support MUTE, SETUP would then fail on the other tracks, and the client could disable the user interface to switch languages.

To enable negotiation with the Require as outlined above, all servers implementing this specification MUST support the "mute" option by adding the "Supported: mute" header to any request it receives with "Require: mute". Clients SHOULD NOT fail to operate with servers that do not implement MUTE.
References


Authors’ Addresses

Aravind Narasimhan
Sun Microsystems, Inc.
901 San Antonio Road
M/S UMPK15-215
Palo Alto, CA 94303
US

Phone: +1 212 558 2494
EMail: Aravind.Narasimhan@Sun.COM

Jonathan Sergent
Sun Microsystems, Inc.
901 San Antonio Road
M/S UMPK15-215
Palo Alto, CA 94303
US

Phone: +1 650 352 6495
EMail: Jonathan.Sergent@Sun.COM
Appendix A. Open issues

1. Should acknowledge RTSP telecon participants.


5. This doesn’t work with multiple queued play requests. There has been some doubt as to whether there are interoperating implementations of queued play, so it may be removed from RTSP. If queued play is not removed from RTSP, this specification will need to support it.