

WebRTC MediaStream Identification in the Session Description Protocol  
draft-ietf-mmusic-msid-17

## Abstract

This document specifies a Session Description Protocol (SDP) Grouping mechanism for RTP media streams that can be used to specify relations between media streams.

This mechanism is used to signal the association between the SDP concept of "media description" and the WebRTC concept of "MediaStream" / "MediaStreamTrack" using SDP signaling.

This document is a work item of the MMUSIC WG, whose discussion list is [mmusic@ietf.org](mailto:mmusic@ietf.org).

## Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](#) [[RFC2119](#)].

## 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 June 16, 2019.

## Copyright Notice

Copyright (c) 2018 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 . . . . .	3
1.1.	Terminology . . . . .	3
1.2.	Structure Of This Document . . . . .	3
1.3.	Why A New Mechanism Is Needed . . . . .	4
1.4.	The WEBRTC MediaStream . . . . .	4
2.	The Msid Mechanism . . . . .	5
3.	Procedures . . . . .	6
3.1.	Handling of non-signalled tracks . . . . .	7
3.2.	Detailed Offer/Answer Procedures . . . . .	9
3.2.1.	Generating the initial offer . . . . .	9
3.2.2.	Answerer processing of the Offer . . . . .	9
3.2.3.	Generating the answer . . . . .	10
3.2.4.	Offerer processing of the answer . . . . .	10
3.2.5.	Modifying the session . . . . .	10
3.3.	Example SDP description . . . . .	10
4.	IANA Considerations . . . . .	11
4.1.	Attribute registration in existing registries . . . . .	11
5.	Security Considerations . . . . .	12
6.	Acknowledgements . . . . .	12
7.	References . . . . .	12
7.1.	Normative References . . . . .	12
7.2.	Informative References . . . . .	13
Appendix A.	Design considerations, rejected alternatives . . . . .	14
Appendix B.	Change log . . . . .	14
B.1.	Changes from alvestrand-rtcweb-msid-00 to -01 . . . . .	14
B.2.	Changes from alvestrand-rtcweb-msid-01 to -02 . . . . .	15
B.3.	Changes from alvestrand-rtcweb-msid-02 to mmusic-msid-00 . . . . .	15
B.4.	Changes from alvestrand-mmusic-msid-00 to -01 . . . . .	15
B.5.	Changes from alvestrand-mmusic-msid-01 to -02 . . . . .	15
B.6.	Changes from alvestrand-mmusic-msid-02 to ietf-mmusic-00 . . . . .	15
B.7.	Changes from mmusic-msid-00 to -01 . . . . .	16

B.8.	Changes from mmusic-msid-01 to -02 . . . . .	16
B.9.	Changes from mmusic-msid-02 to -03 . . . . .	16
B.10.	Changes from mmusic-msid-03 to -04 . . . . .	16
B.11.	Changes from -04 to -05 . . . . .	16
B.12.	Changes from -05 to -06 . . . . .	16
B.13.	Changes from -06 to -07 . . . . .	17
B.14.	Changes from -07 to -08 . . . . .	17
B.15.	Changes from -08 to -09 . . . . .	17
B.16.	Changes from -09 to -10 . . . . .	18
B.17.	Changes from -10 to -11 . . . . .	18
B.18.	Changes from -11 to -12 . . . . .	18
B.19.	Changes from -12 to -13 . . . . .	18
B.20.	Changes from -13 to -14 . . . . .	18
B.21.	Changes from -14 to -15 . . . . .	18
B.22.	Changes from -15 to -16 . . . . .	19
B.23.	Changes from -16 to -17 . . . . .	19
Author's Address	. . . . .	19

## 1. Introduction

### 1.1. Terminology

This document uses terminology from [[I-D.ietf-rtcweb-overview](#)]. In addition, the following terms are used as described below:

**RTP stream** Defined in [[RFC7656](#)] as a stream of RTP packets containing media data.

**MediaStream** Defined in [[W3C.CR-mediacapture-streams-20160519](#)] as an assembly of MediaStreamTracks. One MediaStream can contain multiple MediaStreamTracks, of the same or different types.

**MediaStreamTrack** Defined in [[W3C.CR-mediacapture-streams-20160519](#)] as an unidirectional flow of media data (either audio or video, but not both). Corresponds to the [[RFC7656](#)] term "Source Stream". One MediaStreamTrack can be present in zero, one or multiple MediaStreams.

**Media description** Defined in [[RFC4566](#)] as a set of fields starting with an "m=" field and terminated by either the next "m=" field or by the end of the session description.

### 1.2. Structure Of This Document

This document adds a new Session Description Protocol (SDP) [[RFC4566](#)] mechanism that can attach identifiers to the RTP streams and attaching identifiers to the groupings they form. It is designed for use with WebRTC [[I-D.ietf-rtcweb-overview](#)].

[Section 1.3](#) gives the background on why a new mechanism is needed.

[Section 2](#) gives the definition of the new mechanism.

[Section 3](#) gives the necessary semantic information and procedures for using the msid attribute to signal the association of MediaStreamTracks to MediaStreams in support of the WebRTC API [[W3C.WD-webrtc-20160531](#)].

### 1.3. Why A New Mechanism Is Needed

When media is carried by RTP [[RFC3550](#)], each RTP stream is distinguished inside an RTP session by its SSRC; each RTP session is distinguished from all other RTP sessions by being on a different transport association (strictly speaking, 2 transport associations, one used for RTP and one used for RTCP, unless RTP/RTCP multiplexing [[RFC5761](#)] is used).

SDP [[RFC4566](#)] gives a format for describing an SDP session that can contain multiple media descriptions. According to the model used in [[I-D.ietf-rtcweb-jsep](#)], each media description describes exactly one media source, and if multiple media sources are carried in an RTP session, this is signalled using BUNDLE [[I-D.ietf-mmusic-sdp-bundle-negotiation](#)]; if BUNDLE is not used, each media source is carried in its own RTP session.

The SDP grouping framework [[RFC5888](#)] can be used to group media descriptions. However, for the use case of WebRTC, there is the need for an application to specify some application-level information about the association between the media description and the group. This is not possible using the SDP grouping framework.

### 1.4. The WEBRTC MediaStream

The W3C WebRTC API specification [[W3C.WD-webrtc-20160531](#)] specifies that communication between WebRTC entities is done via MediaStreams, which contain MediaStreamTracks. A MediaStreamTrack is generally carried using a single SSRC in an RTP session (forming an RTP stream. The collision of terminology is unfortunate.) There might possibly be additional SSRCs, possibly within additional RTP sessions, in order to support functionality like forward error correction or simulcast. These additional SSRCs are not affected by this specification.

MediaStreamTracks are unidirectional; they carry media on one direction only.

In the RTP specification, RTP streams are identified using the SSRC field. Streams are grouped into RTP Sessions, and also carry a CNAME. Neither CNAME nor RTP session correspond to a MediaStream. Therefore, the association of an RTP stream to MediaStreams need to be explicitly signaled.

WebRTC defines a mapping (documented in [I-D.ietf-rtcweb-jsep]) where one SDP media description is used to describe each MediaStreamTrack, and the BUNDLE mechanism [I-D.ietf-mmusic-sdp-bundle-negotiation] is used to group MediaStreamTracks into RTP sessions. Therefore, the need is to specify the ID of a MediaStreamTrack and its associated MediaStream for each media description, which can be accomplished with a media-level SDP attribute.

This usage is described in [Section 3](#).

## 2. The Msid Mechanism

This document defines a new SDP [RFC4566] media-level "msid" attribute. This new attribute allows endpoints to associate RTP streams that are described in different media descriptions with the same MediaStreams as defined in [W3C.WD-webrtc-20160531], and to carry an identifier for each MediaStreamTrack in its "appdata" field.

The value of the "msid" attribute consists of an identifier and an optional "appdata" field.

The name of the attribute is "msid".

The value of the attribute is specified by the following ABNF [RFC5234] grammar:

```
msid-value = msid-id [ SP msid-appdata ]
msid-id    = 1*64token-char ; see RFC 4566
msid-appdata = 1*64token-char ; see RFC 4566
```

An example msid value for a group with the identifier "examplefoo" and application data "examplebar" might look like this:

```
msid:examplefoo examplebar
```

The identifier is a string of ASCII characters that are legal in a "token", consisting of between 1 and 64 characters.

Application data (msid-appdata) is carried on the same line as the identifier, separated from the identifier by a space.

The identifier (msid-id) uniquely identifies a group within the scope of an SDP description.

There may be multiple msid attributes in a single media description. This represents the case where a single MediaStreamTrack is present in multiple MediaStreams; the value of "msid-appdata" MUST be identical for all occurrences.

Multiple media descriptions with the same value for msid-id and msid-appdata are not permitted.

Endpoints can update the associations between RTP streams as expressed by msid attributes at any time.

The msid attributes depend on the association of RTP streams with media descriptions, but does not depend on the association of RTP streams with RTP transports; therefore, its mux category (as defined in [I-D.ietf-mmusic-sdp-mux-attributes]) is NORMAL - the process of deciding on MSID attributes doesn't have to take into consideration whether the RTP streams are bundled or not.

### 3. Procedures

This section describes the procedures for associating media descriptions representing MediaStreamTracks within MediaStreams as defined in [W3C.WD-webrtc-20160531].

In the Javascript API described in that specification, each MediaStream and MediaStreamTrack has an "id" attribute, which is a DOMString.

The value of the "msid-id" field in the msid consists of the "id" attribute of a MediaStream, as defined in the MediaStream's WebIDL specification. The special value "-" indicates "no MediaStream".

The value of the "msid-appdata" field in the msid, if present, consists of the "id" attribute of a MediaStreamTrack, as defined in the MediaStreamTrack's WebIDL specification.

When an SDP session description is updated, a specific "msid-id" value continues to refer to the same MediaStream, and a specific "msid-appdata" to the same MediaStreamTrack. There is no memory apart from the currently valid SDP descriptions; if an msid "identifier" value disappears from the SDP and appears in a later negotiation, it will be taken to refer to a new MediaStream.

If the MSID attribute does not conform to the ABNF given here, it SHOULD be ignored.

The following is a high level description of the rules for handling SDP updates. Detailed procedures are in [Section 3.2](#).

- o When a new msid "identifier" value occurs in a session description, and it is not "-", the recipient can signal to its application that a new MediaStream has been added.
- o When a session description is updated to have media descriptions with an msid "identifier" value, with one or more different "appdata" values, the recipient can signal to its application that new MediaStreamTracks have been added, and which MediaStream it has been added to. This is done for each different msid "identifier" value, including the special value "-", which indicates that a MediaStreamTrack has been added with no corresponding MediaStream.
- o If an msid "identifier" value with no "appdata" value appears, it means that the sender did not inform the recipient of the desired identifier of the MediaStreamTrack, and the recipient will assign the "id" value of the created MediaStreamTrack on its own. All msid in a media section that do not have an "appdata" value are assumed to refer to the same MediaStreamTrack.
- o When a session description is updated to no longer list any msid attribute on a specific media description, the recipient can signal to its application that the corresponding MediaStreamTrack has ended.

In addition to signaling that the track is ended when its msid attribute disappears from the SDP, the track will also be signaled as being ended when all associated SSRCS have disappeared by the rules of [\[RFC3550\] section 6.3.4](#) (BYE packet received) and 6.3.5 (timeout), or when the corresponding media description is disabled by setting the port number to zero. Changing the direction of the media description (by setting "sendonly", "recvonly" or "inactive" attributes) will not end the MediaStreamTrack.

The association between SSRCS and media descriptions is specified in [\[I-D.ietf-rtcweb-jsep\]](#).

### 3.1. Handling of non-signalled tracks

Entities that do not use msid will not send msid. This means that there will be some incoming RTP packets that the recipient has no predefined MediaStream id value for.

Note that this handling is triggered by incoming RTP packets, not by SDP negotiation.

When MSID is used, the only time this can happen is when, after the initial negotiation, a negotiation is performed where the answerer adds a `MediaStreamTrack` to an already established connection and starts sending data before the answer is received by the offerer. For initial negotiation, packets won't flow until the ICE candidates and fingerprints have been exchanged, so this is not an issue.

The recipient of those packets will perform the following steps:

- o When RTP packets are initially received, it will create an appropriate `MediaStreamTrack` based on the type of the media (carried in `PayloadType`), and use the MID RTP header extension [[I-D.ietf-mmusic-sdp-bundle-negotiation](#)] (if present) to associate the RTP packets with a specific media section.
- o If the connection is not in the `RTCSignalingState` "stable", it will wait at this point.
- o When the connection is in the `RTCSignalingState` "stable", it will assign ID values.

The following steps are performed to assign ID values:

- o If there is an `msid` attribute, it will use that attribute to populate the "id" field of the `MediaStreamTrack` and associated `MediaStreams`, as described above.
- o If there is no `msid` attribute, the identifier of the `MediaStreamTrack` will be set to a randomly generated string, and it will be signalled as being part of a `MediaStream` with the WebIDL "label" attribute set to "Non-WebRTC stream".
- o After deciding on the "id" field to be applied to the `MediaStreamTrack`, the track will be signalled to the user.

The process above may involve a considerable amount of buffering before the stable state is entered. If the implementation wishes to limit this buffering, it MUST signal to the user that media has been discarded.

It follows from the above that `MediaStreamTracks` in the "default" `MediaStream` cannot be closed by removing the `msid` attribute; the application must instead signal these as closed when the SSRC disappears according to the rules of [RFC 3550 section 6.3.4](#) and 6.3.5 or by disabling the media description by setting its port to zero.



### 3.2. Detailed Offer/Answer Procedures

These procedures are given in terms of [RFC 3264](#)-recommended sections. They describe the actions to be taken in terms of `MediaStreams` and `MediaStreamTracks`; they do not include event signalling inside the application, which is described in JSEP.

#### 3.2.1. Generating the initial offer

For each media description in the offer, if there is an associated outgoing `MediaStreamTrack`, the offerer adds one `"a=msid"` attribute to the section for each `MediaStream` with which the `MediaStreamTrack` is associated. The `"identifier"` field of the attribute is set to the WebIDL `"id"` attribute of the `MediaStream`. If the sender wishes to signal identifiers for the `MediaStreamTracks`, the `"appdata"` field is set to the WebIDL `"id"` attribute of the `MediaStreamTrack`; otherwise it is omitted.

#### 3.2.2. Answerer processing of the Offer

For each media description in the offer, and for each `"a=msid"` attribute in the media description, the receiver of the offer will perform the following steps:

- o Extract the `"appdata"` field of the `"a=msid"` attribute, if present.
- o If the `"appdata"` field exists: Check if a `MediaStreamTrack` with the same WebIDL `"id"` attribute as the `"appdata"` field already exists, and is not in the `"ended"` state. If it is not found, create it.
- o If the `"appdata"` field does not exist, and a `MediaStreamTrack` is not associated with this media section, create one and associate it with this media section for future use.
- o Extract the `"identifier"` field of the `"a=msid"` attribute.
- o Check if a `MediaStream` with the same WebIDL `"id"` attribute already exists. If not, create it.
- o Add the `MediaStreamTrack` to the `MediaStream`
- o Signal to the user that a new `MediaStreamTrack` is available.

### 3.2.3. Generating the answer

The answer is generated in exactly the same manner as the offer. "a=msid" values in the offer do not influence the answer.

### 3.2.4. Offerer processing of the answer

The answer is processed in exactly the same manner as the offer.

### 3.2.5. Modifying the session

On subsequent exchanges, precisely the same procedure as for the initial offer/answer is followed, but with one additional step in the parsing of the offer and answer:

- o For each `MediaStreamTrack` that has been created as a result of previous offer/answer exchanges, and is not in the "ended" state, check to see if there is still an "a=msid" attribute in the present SDP whose "appdata" field is the same as the WebIDL "id" attribute of the track.
- o If no such attribute is found, stop the `MediaStreamTrack`. This will set its state to "ended".

## 3.3. Example SDP description

The following SDP description shows the representation of a WebRTC `PeerConnection` with two `MediaStreams`, each of which has one audio and one video track. Only the parts relevant to the MSID are shown.

Line wrapping, empty lines and comments are added for clarity. They are not part of the SDP.

```
# First MediaStream - id is 4701...
m=audio 56500 UDP/TLS/RTP/SAVPF 96 0 8 97 98
a=msid:47017fee-b6c1-4162-929c-a25110252400
    f83006c5-a0ff-4e0a-9ed9-d3e6747be7d9

m=video 56502 UDP/TLS/RTP/SAVPF 100 101
a=msid:47017fee-b6c1-4162-929c-a25110252400
    b47bdb4a-5db8-49b5-bcdc-e0c9a23172e0

# Second MediaStream - id is 6131....
m=audio 56503 UDP/TLS/RTP/SAVPF 96 0 8 97 98
a=msid:61317484-2ed4-49d7-9eb7-1414322a7aae
    b94006c5-cade-4e0a-9ed9-d3e6747be7d9

m=video 56504 UDP/TLS/RTP/SAVPF 100 101
a=msid:61317484-2ed4-49d7-9eb7-1414322a7aae
    f30bdb4a-1497-49b5-3198-e0c9a23172e0
```

## 4. IANA Considerations

### 4.1. Attribute registration in existing registries

This document requests IANA to register the "msid" attribute in the "att-field (media level only)" registry within the SDP parameters registry, according to the procedures of [\[RFC4566\]](#)

The required information for "msid" is:

- o Contact name, email: IETF, contacted via [mmusic@ietf.org](mailto:mmusic@ietf.org), or a successor address designated by IESG
- o Attribute name: msid
- o Long-form attribute name: MediaStream group Identifier
- o Subject to charset: The attribute value contains only ASCII characters, and is therefore not subject to the charset attribute.
- o Purpose: The attribute can be used to signal the relationship between a WebRTC MediaStream and a set of media descriptions.
- o Appropriate values: The details of appropriate values are given in RFC XXXX.
- o MUX category: NORMAL

The MUX category is defined in [[I-D.ietf-mmusic-sdp-mux-attributes](#)].

## 5. Security Considerations

An adversary with the ability to modify SDP descriptions has the ability to switch around tracks between MediaStreams. This is a special case of the general security consideration that modification of SDP descriptions needs to be confined to entities trusted by the application.

If implementing buffering as mentioned in [Section 3.1](#), the amount of buffering should be limited to avoid memory exhaustion attacks.

Careless generation of identifiers can leak privacy-sensitive information. [[W3C.CR-mediacapture-streams-20160519](#)] recommends that identifiers are generated using UUID class 3 or 4 as a basis, which avoids such leakage.

No other attacks have been identified that depend on this mechanism.

## 6. Acknowledgements

This note is based on sketches from, among others, Justin Uberti and Cullen Jennings.

Special thanks to Flemming Andreassen, Ben Campbell, Miguel Garcia, Martin Thomson, Ted Hardie, Adam Roach, Magnus Westerlund, Alissa Cooper, Sue Hares and Paul Kyzivat for their work in reviewing this draft, with many specific language suggestions.

## 7. References

### 7.1. Normative References

- [[I-D.ietf-mmusic-sdp-mux-attributes](#)]  
Nandakumar, S., "A Framework for SDP Attributes when Multiplexing", [draft-ietf-mmusic-sdp-mux-attributes-12](#) (work in progress), January 2016.
- [[I-D.ietf-rtcweb-jsep](#)]  
Uberti, J., Jennings, C., and E. Rescorla, "Javascript Session Establishment Protocol", [draft-ietf-rtcweb-jsep-14](#) (work in progress), March 2016.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997,  
<<http://www.rfc-editor.org/info/rfc2119>>.

- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, [RFC 3550](#), DOI 10.17487/RFC3550, July 2003, <<http://www.rfc-editor.org/info/rfc3550>>.
- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", [RFC 4566](#), DOI 10.17487/RFC4566, July 2006, <<http://www.rfc-editor.org/info/rfc4566>>.
- [RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, [RFC 5234](#), DOI 10.17487/RFC5234, January 2008, <<http://www.rfc-editor.org/info/rfc5234>>.
- [W3C.CR-mediacapture-streams-20160519]  
Burnett, D., Bergkvist, A., Jennings, C., Narayanan, A., and B. Aboba, "Media Capture and Streams", World Wide Web Consortium CR CR-mediacapture-streams-20160519, May 2016, <<http://www.w3.org/TR/2016/CR-mediacapture-streams-20160519>>.
- [W3C.WD-webrtc-20160531]  
Wilson, C. and J. Kalliokoski, "WebRTC 1.0: Real-time Communication Between Browsers", World Wide Web Consortium WD WD-webrtc-20160531, May 2016, <<http://www.w3.org/TR/2016/WD-webrtc-20160531>>.

## 7.2. Informative References

- [I-D.ietf-mmusic-sdp-bundle-negotiation]  
Holmberg, C., Alvestrand, H., and C. Jennings, "Negotiating Media Multiplexing Using the Session Description Protocol (SDP)", [draft-ietf-mmusic-sdp-bundle-negotiation-29](#) (work in progress), April 2016.
- [I-D.ietf-rtcweb-overview]  
Alvestrand, H., "Overview: Real Time Protocols for Browser-based Applications", [draft-ietf-rtcweb-overview-15](#) (work in progress), January 2016.
- [RFC5761] Perkins, C. and M. Westerlund, "Multiplexing RTP Data and Control Packets on a Single Port", [RFC 5761](#), DOI 10.17487/RFC5761, April 2010, <<http://www.rfc-editor.org/info/rfc5761>>.

- [RFC5888] Camarillo, G. and H. Schulzrinne, "The Session Description Protocol (SDP) Grouping Framework", [RFC 5888](#), DOI 10.17487/RFC5888, June 2010, <<http://www.rfc-editor.org/info/rfc5888>>.
- [RFC7656] Lennox, J., Gross, K., Nandakumar, S., Salgueiro, G., and B. Burman, Ed., "A Taxonomy of Semantics and Mechanisms for Real-Time Transport Protocol (RTP) Sources", [RFC 7656](#), DOI 10.17487/RFC7656, November 2015, <<http://www.rfc-editor.org/info/rfc7656>>.

#### [Appendix A.](#) Design considerations, rejected alternatives

One suggested mechanism has been to use CNAME instead of a new attribute. This was abandoned because CNAME identifies a synchronization context; one can imagine both wanting to have tracks from the same synchronization context in multiple MediaStreams and wanting to have tracks from multiple synchronization contexts within one MediaStream (but the latter is impossible, since a MediaStream is defined to impose synchronization on its members).

Another suggestion has been to put the msid value within an attribute of RTCP SR (sender report) packets. This doesn't offer the ability to know that you have seen all the tracks currently configured for a MediaStream.

A suggestion that survived for a number of drafts was to define "msid" as a generic mechanism, where the particular semantics of this usage of the mechanism would be defined by an "a=wms-semantic" attribute. This was removed in April 2015.

#### [Appendix B.](#) Change log

This appendix should be deleted before publication as an RFC.

##### [B.1.](#) Changes from alvestrand-rtcweb-msid-00 to -01

Added track identifier.

Added inclusion-by-reference of [draft-lennox-mmusic-source-selection](#) for track muting.

Some rewording.

## B.2. Changes from alvestrand-rtcweb-msid-01 to -02

Split document into sections describing a generic grouping mechanism and sections describing the application of this grouping mechanism to the WebRTC MediaStream concept.

Removed the mechanism for muting tracks, since this is not central to the MSID mechanism.

## B.3. Changes from alvestrand-rtcweb-msid-02 to mmusic-msid-00

Changed the draft name according to the wishes of the MMUSIC group chairs.

Added text indicting cases where it's appropriate to have the same appdata for multiple SSRCs.

Minor textual updates.

## B.4. Changes from alvestrand-mmusic-msid-00 to -01

Increased the amount of explanatory text, much based on a review by Miguel Garcia.

Removed references to BUNDLE, since that spec is under active discussion.

Removed distinguished values of the MSID identifier.

## B.5. Changes from alvestrand-mmusic-msid-01 to -02

Changed the order of the "msid-semantic: " attribute's value fields and allowed multiple identifiers. This makes the attribute useful as a marker for "I understand this semantic".

Changed the syntax for "identifier" and "appdata" to be "token".

Changed the registry for the "msid-semantic" attribute values to be a new registry, based on advice given in Atlanta.

## B.6. Changes from alvestrand-mmusic-msid-02 to ietf-mmusic-00

Updated terminology to refer to m-lines rather than RTP sessions when discussing SDP formats and the ability of other linking mechanisms to refer to SSRCs.

Changed the "default" mechanism to return independent streams after considering the synchronization problem.

Removed the space from between "msid-semantic" and its value, to be consistent with [RFC 5576](#).

**B.7.** Changes from mmusic-msid-00 to -01

Reworked msid mechanism to be a per-m-line attribute, to align with [draft-roach-mmusic-unified-plan](#).

**B.8.** Changes from mmusic-msid-01 to -02

Corrected several missed cases where the word "ssrc" was not changed to "M-line".

Added pointer to unified-plan (which should be moved to point to -jsep)

Removed suggestion that ssrc-group attributes can be used with "msid-semantic", it is now only the msid-semantic registry.

**B.9.** Changes from mmusic-msid-02 to -03

Corrected even more cases where the word "ssrc" was not changed to "M-line".

Added the functionality of using an asterisk (\*) in the msid-semantic line, in order to remove the need for listing all msids in the msid-semantic line when only one msid-semantic is in use.

Removed some now-unnecessary text.

**B.10.** Changes from mmusic-msid-03 to -04

Changed title to reflect focus on WebRTC MediaStreams

Added a section on receiver-side media stream control, using the "msid-control" attribute.

**B.11.** Changes from -04 to -05

Removed the msid-control section after WG discussion.

Removed some text that seemed only to pertain to resolved issues.

**B.12.** Changes from -05 to -06

Addressed issues found in Fleming Andreassen's review

Referenced JSEP rather than unified-plan for the M-line mapping model



Relaxed MSID definition to allow "token-char" in values rather than a-z 0-9 hyphen; tightened ABNF by adding length description to it.

Deleted discussion of abandoned alternatives, as part of preparing for publication.

Added a "detailed procedures" section to the WMS semantics description.

Added IANA registration of the "msid-semantic" attribute.

#### B.13. Changes from -06 to -07

Changed terminology from referring to "WebRTC device" to referring to "entities that implement the WMS semantic".

Changed names for ABNF constructions based on a proposal by Paul Kyzivat.

Included a section on generic offer/answer semantics.

#### B.14. Changes from -07 to -08

Removed [Appendix B](#) that described the (now obsolete) ssrc-specific usage of MSID.

Adopted a restructuring of the IANA section based on a suggestion from Martin Thomson.

A number of text and ABNF clarifications based on suggestions from Ted Hardie, Paul Kyzivat and Adam Roach.

Changed the "non-signalled track handling" to create a single stream with multiple tracks again, according to discussions at TPAC in November 2014

#### B.15. Changes from -08 to -09

Removed "wms-semantic" and all mention of multiple semantics for msid, as agreed at the Dallas IETF, March 2015.

Addressed a number of review comments from Fleming Andresen and others.

Changed the term "m-line" to "media description", since that is the term used in [RFC 4566](#).

Tried to make sure this document does not describe the API to the application.

**B.16.** Changes from -09 to -10

Addressed review comments from Paul Kyzivat.

**B.17.** Changes from -10 to -11

Defined the semantics of multiple MSIDs in a media section to be a `MediaStreamTrack` present in multiple `MediaStreams`.

Made an explicit note that `MediaStreamTracks` are unidirectional.

Disallowed the option of sending multiple media sections with the same `msid` (id and `appdata` identical).

**B.18.** Changes from -11 to -12

Added `mux-category` to the IANA considerations section.

**B.19.** Changes from -12 to -13

Modified registration description to delete dependency on -4566-bis

**B.20.** Changes from -13 to -14

Addressed nits found in Gen-ART review

**B.21.** Changes from -14 to -15

Added the terminology section. Switched from "(RTP) media stream" to "RTP stream" per [RFC 7656](#).

Added a mention of random ID generation to the security considerations section.

Moved definition pointers for `MediaStream` and `MediaStreamTrack` to the "mediacapture-streams" document.

Added note that syntactically invalid MSID fields SHOULD be ignored.

Various small changes based on review feedback during IESG processing.

**B.22.** Changes from -15 to -16

Added the special "-" value that means "no MediaStream".

Changed instances of a MediaStreamTrack being "closed" to saying it's "ended", in accordance with WebRTC terminology.

**B.23.** Changes from -16 to -17

Added text to allow omitting track identifiers, per JSEP PR #850

**Author's Address**

Harald Alvestrand  
Google  
Kungsbron 2  
Stockholm 11122  
Sweden

Email: [harald@alvestrand.no](mailto:harald@alvestrand.no)