

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 15, 2015

H. Alvestrand
Google
February 11, 2015

WebRTC MediaStream Identification in the Session Description Protocol
draft-ietf-mmusic-msid-08

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 "m-line" 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.

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 August 15, 2015.

Copyright Notice

Copyright (c) 2015 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. Structure Of This Document	3
1.2. Why A New Mechanism Is Needed	3
1.3. Application to the WEBRTC MediaStream	4
2. The Msid Mechanism	5
3. The Msid-Semantic Attribute	6
4. Generic SDP Offer/Answer Procedures	7
4.1. Generating the Initial Offer	7
4.2. Answerer Processing of the Offer	7
4.3. Generating the Answer	7
4.4. Offerer Processing of the Answer	7
5. Applying Msid to WebRTC MediaStreams	7
5.1. Handling of non-signalled tracks	9
5.2. Detailed Offer/Answer Procedures	10
5.2.1. Generating the initial offer	10
5.2.2. Parsing the initial offer	10
5.2.3. Generating the answer	11
5.2.4. Offerer processing of the answer	11
5.2.5. Modifying the session	11
6. IANA Considerations	11
6.1. Attribute registration in existing registries	11
6.2. New registry creation	12
7. Security Considerations	13
8. Acknowledgements	13
9. References	13
9.1. Normative References	13
9.2. Informative References	14
Appendix A. Design considerations, rejected alternatives	14
Appendix B. Change log	14
B.1. Changes from alvestrand-rtcweb-msid-00 to -01	15
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	16
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	17
B.12.	Changes from -05 to -06	17
B.13.	Changes from -06 to -07	17
B.14.	Changes from -07 to -08	17
Author's Address	18

1. Introduction

1.1. Structure Of This Document

This document adds a new Session Description Protocol (SDP) [RFC4566] mechanism that can associate application layer identifiers with the binding between media streams, attaching identifiers to the media streams and attaching identifiers to the groupings they form.

Section 1.2 gives the background on why a new mechanism is needed.

Section 2 gives the definition of the new mechanism.

Section 3 gives the definition of the msid-semantic field, which gives the possibility of using MSIDs with different semantics in the same SDP message.

Section 5 gives the application of the new mechanism for providing necessary semantic information for the association of `MediaStreamTracks` to `MediaStreams` in the WebRTC API [W3C.WD-webrtc-20120209].

1.2. Why A New Mechanism Is Needed

When media is carried by RTP [RFC3550], each RTP media 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 gives a description based on m-lines. According to the model used in [I-D.ietf-rtcweb-jsep], each m-line 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.

There exist cases where an application using RTP and SDP needs to signal some relationship between RTP media streams that may be carried in either the same RTP session or different RTP sessions. For instance, there may be a need to signal a relationship between a video track and an audio track, and where the generator of the SDP does not yet know if they will be carried in the same RTP session or different RTP sessions.

The SDP grouping framework [RFC5888] can be used to group m-lines. However, there is sometimes the need for an application to specify some application-level information about the association between the m-line and the group. This is not possible using the SDP grouping framework.

1.3. Application to the WEBRTC MediaStream

The W3C WebRTC API specification [W3C.WD-webrtc-20120209] 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 media 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. This complication is ignored below.

In the RTP specification, media 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 media stream to MediaStreams need to be explicitly signaled.

WebRTC defines a mapping (documented in [I-D.ietf-rtcweb-jsep]) where one SDP m-line is used to describe each MediaStreamTrack, and that 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 m-line, which can be accomplished with a media-level SDP attribute.

This usage is described in Section 5.

2. The Msid Mechanism

This document defines a new SDP [RFC4566] media-level "msid" attribute. This new attribute allows endpoints to associate RTP media streams that are carried in the same or different m-lines. The attribute also allows application-specific information to the association.

The value of the "msid" attribute consists of an identifier and optional application-specific data.

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. It MUST be unique among the identifier values used in the same SDP session. It is RECOMMENDED that it is generated using a random-number generator.

Application data is carried on the same line as the identifier, separated from the identifier by a space.

The identifier uniquely identifies a group within the scope of an SDP description.

There may be multiple msid attributes in a single media description. There may also be multiple media descriptions that have the same value for identifier and application data.

Endpoints can update the associations between RTP media streams as expressed by msid attributes at any time; the semantics and restrictions of such grouping and ungrouping are application dependent.

3. The Msid-Semantic Attribute

A session-level attribute is defined for signaling the semantics associated with an msid grouping. This allows msid groupings with different semantics to coexist.

This OPTIONAL attribute gives the group identifier and its group semantic; it carries the same meaning as the ssrc-group-attr of RFC 5576 section 4.2, but uses the identifier of the group rather than a list of SSRC values.

This attribute MUST be present if "a=msid" is used.

An empty list of identifiers is an indication that the sender supports the indicated semantic, but has no msid groupings of the given type in the present SDP.

An identifier of "*" is an indication that all "a=msid" lines in the SDP have this specific semantic. If "*" is not used, each msid-id in the SDP MUST appear in one and only one "msid-semantic" line.

The name of the attribute is "msid-semantic".

The value of the attribute is given by the following ABNF:

```
msid-semantic-value = msid-semantic msid-list
msid-semantic = token ; see RFC 4566
msid-list = *(" " msid-id) / " *"
```

The semantic field holds values from the IANA registry "Semantics for the msid-semantic SDP attribute" (which is defined in Section 6).

An example msid-semantic might look like this, if a semantic LS was registered by IANA for the same purpose as the existing LS grouping semantic:

```
a=msid-semantic:LS xyzzy forolow
```

This means that the SDP description has two lip sync groups, with the group identifiers xyzzy and forolow, respectively.

The msid-semantic attribute can occur more than once, but MUST NOT occur more than once with the same msid-semantic value.

4. Generic SDP Offer/Answer Procedures

In accordance with guidance on definitions of SDP extensions, this section gives the generic procedures that have to be followed by all implementations of Msid, independent of which semantics they support.

Note that the use of msid is not negotiated; each side declares what semantics it uses. This means that an offerer has to be willing and able to take appropriate action if the other side does not wish to use the semantic, and an answerer adding new semantics to an answer has to be willing and able to deal with the offerer not wishing to use that semantic.

4.1. Generating the Initial Offer

An entity wishing to use an MSID semantic MUST add one or more "msid-semantic" attributes to its session level attributes, indicating the MSID semantic it wishes to have available..

4.2. Answerer Processing of the Offer

If an "msid-semantic" attribute is present in the offer, and the answerer wishes to use the indicated semantic, the offerer MUST follow the procedures described for that semantic.

4.3. Generating the Answer

An entity wishing to use an MSID semantic MUST add one or more "msid-semantic" attributes to its session level attributes, indicating the MSID semantic it wishes to have available. If the answerer does not wish to use one or more of the semantics indicated in the offer, the answerer MUST NOT include "msid-semantic" lines indicating these semantics in the answer.

4.4. Offerer Processing of the Answer

If an "msid-semantic" attribute is present in the answer, and the offerer wishes to use the indicated semantic, the offerer MUST follow the procedures described for that semantic. The offerer MUST follow the procedures for all semantics that were indicated in its offer and were also present in the answer.

5. Applying Msid to WebRTC MediaStreams

This section creates a new semantic for use with the framework defined in Section 2, to be used for associating m-lines representing MediaStreamTracks within MediaStreams as defined in [W3C.WD-webrtc-20120209].

In the Javascript API, each `MediaStream` and `MediaStreamTrack` has an "id" attribute, which is a `DOMString`.

The semantic token for this semantic is "WMS" (short for WebRTC Media Stream).

The value of the "identifier" field in the msid consists of the "id" attribute of a `MediaStream`, as defined in its WebIDL specification.

The value of the "appdata" field in the msid consists of the "id" attribute of a `MediaStreamTrack`, as defined in its WebIDL specification.

If two different m-lines have MSID attributes with the same value for identifier and appdata, it means that these two m-lines are both intended for the same `MediaStreamTrack`. So far, no semantic for such a mixture have been defined, but this specification does not forbid the practice.

When an SDP description is updated, a specific msid "identifier" continues to refer to the same `MediaStream`, and a specific "appdata" to the same `MediaStreamTrack`. Once negotiation has completed on a session, 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`.

The following are the rules for handling updates of the list of m-lines and their msid values.

- o When a new msid "identifier" value occurs in the description, the recipient can signal to its application that a new `MediaStream` has been added.
- o When a description is updated to have more media sections with the same msid "identifier" value, but different "appdata" values, the recipient can signal to its application that new `MediaStreamTracks` have been added to the `MediaStream`.
- o When a description is updated to no longer list the 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 closed when its msid attribute disappears from the SDP, the track will also be signaled as being closed when all associated SSRCS have disappeared by the rules of [RFC3550] section 6.3.4 (BYE packet received) and 6.3.5 (timeout), and when the corresponding media section is disabled by setting the

port number to zero. Changing the direction of the media section (by setting "sendonly", "recvonly" or "inactive" attributes) will not close the `MediaStreamTrack`.

The association between SSRCs and m-lines is specified in [I-D.ietf-rtcweb-jsep].

5.1. Handling of non-signalled tracks

Entities that do not use the WMS semantic will not send "msid-semantic:WMS". 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.

Handling will depend on whether or not the `msid-semantic:WMS` attribute is present. There are two cases:

- o No "msid-semantic:WMS" attribute is present. The SDP session is assumed to be a backwards-compatible session. All incoming media, on all m-lines that are part of the SDP session, are assumed to belong to tracks of the same media stream (the "default media stream"). The identifier of this media stream and of the media stream track is a randomly generated string; the WebIDL "label" attribute of this media stream will be set to "Non-WMS stream".
- o An "msid-semantic:WMS" attribute is present. In this case, the sender implements the WMS semantic, and the packets are either caused by a bug or by timing skew between the arrival of the media packets and the SDP description. These packets MAY be discarded, or they MAY be buffered for a while in order to allow immediate startup of the media stream when the SDP description is updated. The arrival of media packets MUST NOT cause a new `MediaStreamTrack` to be signaled.

If an entity wishing to use the WMS semantic sends a description, it MUST include the `msid-semantic:WMS` attribute, even if no media streams are sent. This allows us to distinguish between the case of no media streams at the moment and the case of SDP generated by an entity that wishes to use the backwards-compatible mechanism.

It follows from the above that the media receiver implementing the WMS semantic must have the SDP of the other party before it can decide correctly which of the two cases described above applies. RTP media packets that arrive before the remote party's SDP MUST be

buffered or discarded, and MUST NOT cause a new MediaStreamTrack to be signalled.

It follows from the above that media stream tracks in the "default" media stream 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 m-line by setting its port to zero.

5.2. 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.

They are specifically applicable to the WMS semantic; other semantics will have their own consideration.

5.2.1. Generating the initial offer

For each media section in the offer, if there is an associated 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, and the "appdata" field is set to the WebIDL "id" attribute of the MediaStreamTrack.

The offerer adds an "msid-semantic:WMS" field to the session-level headers, and appends to it either a list of all the identifiers used in the offer, or the single character "*".

5.2.2. Parsing the initial offer

For each media section in the offer, and for each "a=msid" attribute in the media section where the "msid-id" is associated with the "WMS" semantic, the receiver of the offer will perform the following steps:

- o Extract the "appdata" field of the "a=msid" attribute
- o 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 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

5.2.3. Generating the answer

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

This includes adding a "msid-semantic:WMS" attribute in the session-level headers, independent of whether or not such a header was present in the offer.

5.2.4. Offerer processing of the answer

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

5.2.5. 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, close the MediaStreamTrack. This will set its state to "ended".

6. IANA Considerations

6.1. 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, or a successor address designated by IESG
- o Attribute name: msid
- o Long-form attribute name: Media stream 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 gives an association over a set of m-lines. For example, it can be used to signal the relationship between a WebRTC MediaStream and a set of m-lines.
- o Appropriate values: The details of appropriate values are given in RFC XXXX.

This document requests IANA to register the "msid-semantic" attribute in the "att-field (session level) registry within the SDP parameters registry, according to the same procedures.

The required information is:

- o Contact name, email: IETF, contacted via mmusic@ietf.org, or a successor address designated by IESG
- o Attribute name: msid-semantic
- o Long-form attribute name: Msid group semantic 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 gives the semantics of an association over a set of m-lines.
- o Appropriate values: The details are given in RFC XXXX.

6.2. New registry creation

This document requests IANA to create a new registry called "Semantics for the msid-semantic SDP attribute" in the "Session Description Protocol (SDP) Parameters" group. This registry operates on the Expert Review policy [RFC5226]. Usage of the registry is expected to be low, so the expert should feel free to consult widely if a new request ever comes in.

This document requests IANA to register the "WMS" semantic within this new registry.

The required information is:

- o Description: WebRTC Media Stream, as given in RFC XXXX.
- o Token: WMS
- o Standards track reference: RFC XXXX

IANA is requested to replace "RFC XXXX" with the RFC number of this document upon publication.

7. Security Considerations

An adversary with the ability to modify SDP descriptions has the ability to switch around tracks between media streams. 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 5.1, the amount of buffering should be limited to avoid memory exhaustion attacks.

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

8. Acknowledgements

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

Special thanks to Flemming Andreassen, Miguel Garcia, Martin Thomson, Ted Hardie, Adam Roach and Paul Kyzivat for their work in reviewing this draft, with many specific language suggestions.

9. References

9.1. Normative References

- [I-D.ietf-rtcweb-jsep]
Uberti, J., Jennings, C., and E. Rescorla, "Javascript Session Establishment Protocol", draft-ietf-rtcweb-jsep-08 (work in progress), October 2014.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.
- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

[W3C.WD-webrtc-20120209]
Bergkvist, A., Burnett, D., Jennings, C., and A. Narayanan, "WebRTC 1.0: Real-time Communication Between Browsers", World Wide Web Consortium WD WD-webrtc-20120209, February 2012, <<http://www.w3.org/TR/2012/WD-webrtc-20120209>>.

9.2. 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-07 (work in progress), April 2014.
- [RFC5761] Perkins, C. and M. Westerlund, "Multiplexing RTP Data and Control Packets on a Single Port", RFC 5761, April 2010.
- [RFC5888] Camarillo, G. and H. Schulzrinne, "The Session Description Protocol (SDP) Grouping Framework", RFC 5888, June 2010.

Appendix A. Design considerations, rejected alternatives

This appendix should be deleted before publication as an RFC.

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 media stream.

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 whne 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

Author's Address

Harald Alvestrand
Google
Kungsbron 2
Stockholm 11122
Sweden

Email: harald@alvestrand.no