

Network Working Group
Internet-Draft
Intended status: Informational
Expires: February 27, 2013

H. Alvestrand
Google
August 26, 2012

Resolution Constraints in Web Real Time Communications
draft-alvestrand-constraints-resolution-00

Abstract

This document specifies the constraints necessary for a Javascript application to successfully indicate to a browser that supports WebRTC what resolutions it desires on a video stream.

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 February 27, 2013.

Copyright Notice

Copyright (c) 2012 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.	Disposition of this text	3
2.	Usage considerations	3
3.	Usage examples	4
3.1.	Examples with GetUserMedia	4
3.2.	Possible SDP mappings	5
4.	IANA Considerations	5
5.	Security Considerations	6
6.	Acknowledgements	6
7.	References	6
7.1.	Normative References	6
7.2.	Informative References	7
	Author's Address	7

1. Introduction

There are a number of scenarios where it's useful for a WebRTC application to indicate to the WebRTC implementation in the supported browser what the desired characteristics of a video stream are.

These include, but are not limited to:

- o Specifying a minimum desired resolution for a given application, in order to control the user experience or resource tradeoffs made by the browser to favour a particular stream
- o Specifying a maximum desired resolution for a given stream, in order to save some resource (bandwidth, CPU...), possibly outside of the browser where the browser can't tell that it's exceeding a constraint
- o Specifying resolutions that are a reasonable fit for the current usage of the video stream, for instance fitting with the number of pixels available on the part of a device's display surface that is devoted to displaying this video stream
- o Specifying the shape of a video stream, in order to fit the video onto a display surface without the need for black bars or image distortion

Similar considerations apply for framerate.

1.1. Disposition of this text

This draft is written in order to get something specific out to refer to during spec-writing and implementation. The text may eventually get merged into either the IETF document on SDP usage by RTCWEB, or the W3C WEBRTC document on PeerConnection.

2. Usage considerations

These constraints are usable in several places:

- o As constraints to the getUserMedia call [[W3C.WD-mediacapture-streams-20120628](#)], where they serve to guide the configuration of the camera obtained, and may influence the choice of camera.
- o As constraints to the addStream call on a PeerConnection [[W3C.WD-webrtc-20120821](#)], where they serve to guide the configuration of the codec that encodes the video content for transmission.

- o As constraints applied to an existing local video stream using the "change constraints" API, where it may cause the video engine to reconfigure the device or codec for that particular stream.
- o As constraints applied to an incoming video stream using the "change constraints" API on a `MediaStreamTrack`, where it serves to inform the video engine about the desirable properties of the video track, which may lead to the video engine choosing to reencode the video and/or signal a remote video source that it wishes certain constraints to be put in place.

All of the constraints may be meaningful in both "mandatory" and "optional" forms.

3. Usage examples

See [Section 4](#) for the actual definition of the constraints used here.

3.1. Examples with `getUserMedia`

A constraint saying that we absolutely must have a minimum resolution of 1024x768:

```
getUserMedia({
  video: { mandatory: { minWidth: 1024, minHeight: 768 } }
}, successCallback, errorCallback);
```

A constraint saying that we'd prefer 60 frames per second, if available, and if we can get that, we'd like to limit the max resolution, but in all cases, the screen must be clamped to a 4:3 aspect ratio - 16:9 or odd aspect ratios are not acceptable to this application:

```
getUserMedia({
  video: {
    mandatory: { minAspectRatio: 1.333, maxAspectRatio: 1.334 },
    optional [
      { minFrameRate: 60 },
      { maxWidth: 640 },
      { maxHeight: 480 }
    ]
  }
}, successCallback, errorCallback);
```


3.2. Possible SDP mappings

This document does not specify or constrain how constraints get reflected into SDP (if they do); that's an implementor decision.

The examples below are thought exercises, based on [\[I-D.lennox-mmusic-sdp-source-selection\]](#) and [\[I-D.alvestrand-rtcweb-resolution\]](#).

An optional constraint has been applied to an incoming stream where both upper and lower are constrained to 320x200. The stream has been assigned to a hardware video decoder that can decode most resolutions up to 1024x768, in any aspect ratio, but only if all divisions are divisible by 4. The incoming stream has SSRC 1234.

Line breaks are added for readability.

```
m=video
a=remote-ssrc:1234 imageattr:* [x=320,y=200,q=1.0] \
    [x=[120:4:1024],y=[100:4:768],q=0.2]
```

4. IANA Considerations

This document requests IANA to register constraints in the "RTCWeb Media Constraints" registry created by [\[I-D.burnett-rtcweb-constraints-registry\]](#). NOTE: The registrations assume that this document is updated to no longer have "video" as part of the name, but have "video" as a field-of-use in the registration.

The definitions of width, height and aspect ratio are taken from [\[RFC6236\]](#).

- o MinWidth - valid for video. Corresponds to the "x" value (pixel count) from [RFC 6236](#). Only integer values are valid.
- o MaxWidth - valid for video. Definition as for MinWidth.
- o MinHeight - valid for video. Corresponds to the "y" value (pixel count) from [RFC 6236](#). Only integer values are valid.
- o MaxHeight - valid for video. Definition as for MinHeight.

- o `MinAspectRatio` - valid for video. Corresponds to the "par" (picture aspect ratio), with "sar" set to 1.0. A 4:3 format display corresponds to an `AspectRatio` of 1.3333. Floating point values are valid.
- o `MaxAspectRatio` - valid for video. Definition as for `MinAspectRatio`.
- o `MinFramerate` - valid for video. Corresponds to the framerate defined in [RFC4566], the "a=framerate" attribute.
- o `MaxFramerate` - valid for video. Definition as for `MinFramerate`.

The contact person is Harald Alvestrand <hta@google.com>.

Change control for the registration is with the IETF, as designated by the IESG.

Note that `MinFramerate` defines a lower bound for the `a=framerate` attribute, which is itself defined as an upper limit; this means that even if a high framerate is negotiated, the actual framerate used may be lower due to temporary considerations (for instance CPU or bandwidth, or simply lack of movement in the picture).

5. Security Considerations

No security considerations particular to these specific constraints have so far been identified.

6. Acknowledgements

Special thanks are given to Dan Burnett, Cullen Jennings, the IETF RTCWEB WG and the W3C WEBRTC WG for strongly influencing this memo.

7. References

7.1. Normative References

- [I-D.burnett-rtcweb-constraints-registry]
Burnett, D., "IANA Registry for RTCWeb Media Constraints", [draft-burnett-rtcweb-constraints-registry-01](#) (work in progress), April 2012.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.

- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", [RFC 4566](#), July 2006.
- [RFC6236] Johansson, I. and K. Jung, "Negotiation of Generic Image Attributes in the Session Description Protocol (SDP)", [RFC 6236](#), May 2011.

[7.2.](#) Informative References

- [I-D.alvestrand-rtcweb-resolution]
Alvestrand, H., "RTCWEB Resolution Negotiation",
[draft-alvestrand-rtcweb-resolution-00](#) (work in progress),
April 2012.
- [I-D.lennox-mmusic-sdp-source-selection]
Lennox, J. and H. Schulzrinne, "Mechanisms for Media Source Selection in the Session Description Protocol (SDP)", [draft-lennox-mmusic-sdp-source-selection-04](#) (work in progress), March 2012.
- [W3C.WD-mediacapture-streams-20120628]
Burnett, D. and A. Narayanan, "Media Capture and Streams",
World Wide Web Consortium WD WD-mediacapture-streams-
20120628, June 2012, <<http://www.w3.org/TR/2012/WD-mediacapture-streams-20120628>>.
- [W3C.WD-webrtc-20120821]
Bergkvist, A., Burnett, D., Jennings, C., and A.
Narayanan, "WebRTC 1.0: Real-time Communication Between Browsers",
World Wide Web Consortium WD WD-webrtc-
20120821, August 2012,
<<http://www.w3.org/TR/2012/WD-webrtc-20120821>>.

Author's Address

Harald Alvestrand
Google

Email: harald@alvestrand.no

