

W3C

WebRTC/MediaCapture

WG Meeting

May 2, 2017
8 AM PDT

Chairs: Stefan Hakansson
Bernard Aboba
Harald Alvestrand (Emeritus)

W3C WG IPR Policy

- This group abides by the W3C patent policy
<https://www.w3.org/Consortium/Patent-Policy-20040205>
- Only people and companies listed at
<https://www.w3.org/2004/01/pp-impl/47318/status> are
allowed to make substantive contributions to the
WebRTC specs

Welcome!

- Welcome to the interim meeting of the W3C WebRTC WG!
- During this meeting, we hope to make progress on outstanding issues within webrtc-pc and mediacapture-main
- Editor's Draft updates to follow meeting

webrtc-pc

- As announced, we are targeting to submit a transition (to CR) request after this meeting!
- Of course we need to deal with open Issues before - hoping this meeting will be fruitful
- Reminder: we would like feedback from implementers on features that are not on their radar for implementation

About this Virtual Meeting

Information on the meeting:

- Meeting info:
 - https://www.w3.org/2011/04/webrtc/wiki/May_2_2017
- Link to latest drafts:
 - <https://rawgit.com/w3c/mediacapture-main/master/getusermedia.html>
 - <https://rawgit.com/w3c/webrtc-pc/master/webrtc.html>
 - <https://rawgit.com/w3c/webrtc-stats/master/webrtc-stats.html>
- Link to Slides has been published on [WG wiki](#)
- Scribe? IRC <http://irc.w3.org/> Channel: [#webrtc](#)
- The meeting is being recorded.
- WebEx info [here](#)

For Discussion Today

- WebRTC-PC

- Pull Requests

- [Issue 1073/Issue 1116/PR 1134](#): Adding more detail about how `getParameters` and `setParameters` work. (Taylor)
 - [Issue 1101/PR 1133](#): `RTCRtpContributingSource` naming (Taylor)
 - [Issue 763/PR 1150](#): Handling of Simulcast Errors (Bernard)
 - [Issue 1092/PR 1115](#): DTLS failures (Bernard)
 - [Issue 1138/PR 1145](#): `FrozenArray`, `sequence` and `SameObject` (foolip)

- Issues

- [Issue 1128](#): Does MSID still work? (Taylor)

- Media Capture

- Pull Requests

- [Issue 434/PR 445](#): Add `autoGainControl` and `noiseSupression` constraints (jan-ivar)

WebRTC PC Pull Requests

- [Issue 1073](#)/[Issue 1116](#)/[PR 1134](#): Adding more detail about how `getParameters` and `setParameters` work. (Taylor)
- [Issue 1101](#)/[PR 1133](#): `RTCRtpContributingSource` naming (Taylor)
- [Issue 763](#)/[PR 1150](#): Handling of Simulcast Errors (Bernard)
- [Issue 1092](#)/[PR 1115](#): DTLS failures (Bernard)
- [Issue 1138](#)/[PR 1145](#): `FrozenArray`, `sequence` and `SameObject` (foolip)

[Issue 1073](#)/[Issue 1116](#)/[PR 1134](#): Adding more detail about how `getParameters` and `setParameters` work (Taylor)

- Decision from April virtual interim: “`sendEncodings`” parameter can be used to preemptively set any read-write encoding parameters.
- Other things specified in PR 1134:
 - Before a remote description is set, “`sender.getParameters`” only returns encoding parameters. Which can be modified by “`setParameters`”.
 - Setting a remote description populates the sender’s codec and header extension parameters.
 - For an `RTCRtpReceiver`, SSRCs come from the remote description and everything else comes from the local description.
 - Receiver’s parameters are entirely determined by SDP, since there is no “`receiver.setParameters`”.
 - This means that if SSRCs are unsignaled, they won’t be returned by “`receiver.getParameters`”.

Issue 1101/PR 1133: RTCRtpContributingSource naming (Taylor)

- As decided at April virtual interim, “getContributingSources” was split into two methods: “getContributingSources” and “getSynchronizationSources”.
- The objects returned by these methods are:

```
interface RTCRtpContributingSource {
  readonly attribute DOMHighResTimeStamp timestamp;
  readonly attribute unsigned long      source;
  readonly attribute byte?              audioLevel;
};

interface RTCRtpSynchronizationSource {
  readonly attribute DOMHighResTimeStamp timestamp;
  readonly attribute unsigned long      source;
  readonly attribute byte               audioLevel;
  readonly attribute boolean?          voiceActivityFlag;
};
```

Issue 763/PR 1150: Handling of Simulcast Errors (Bernard)

- Problem: Encoding parameter errors (in simulcast as well as other uses) difficult to diagnose.
- Proposed fix adds ErrorDetailType values:
 - invalid-encoding-parameters: Invalid RTCRtpEncodingParameters dictionary provided to addTransceiver() or setParameters().
 - invalidParameters[] attribute: set to the parameters that were invalid.
 - too-many-encodings: Too many encodings requested in addTransceiver().
 - maxEncodings attribute: how many encodings are supported. Problem: is this dynamic?
 - invalid-codecs: Invalid or unavailable codecs provided to setParameters() or setCodecPreferences().
 - invalidCodecs[] attribute: set to the mimeType of the codecs that were invalid.

Issue 1092/PR 1115: DTLS Failures (Bernard)

- Goal: To provide information on recoverable DTLS errors (and distinguish them from unrecoverable errors). Examples:

- Cryptographic negotiation failure (43)
- Fingerprint not verified (42)
- Protocol version failure (70)

- Approach:

- Add "dtls-failure" value to RTCErrorDetailType
- Add tlsAlert attribute to RTCError, containing the (D)TLS alert value:

<https://www.iana.org/assignments/tls-parameters/tls-parameters.xhtml#tls-parameters-6>

Value	Description	DTLS-OK	Reference
0	close_notify	Y	[RFC5246]
1-9	Unassigned		
10	unexpected_message	Y	[RFC5246]
11-19	Unassigned		
20	bad_record_mac	Y	[RFC5246]
21	decryption_failed	Y	[RFC5246]
22	record_overflow	Y	[RFC5246]
23-29	Unassigned		
30	decompression_failure	Y	[RFC5246]
31-39	Unassigned		
40	handshake_failure	Y	[RFC5246]
41	no_certificate_RESERVED	Y	[RFC5246]
42	bad_certificate	Y	[RFC5246]
43	unsupported_certificate	Y	[RFC5246]
44	certificate_revoked	Y	[RFC5246]
45	certificate_expired	Y	[RFC5246]
46	certificate_unknown	Y	[RFC5246]
47	illegal_parameter	Y	[RFC5246]
48	unknown_ca	Y	[RFC5246]
49	access_denied	Y	[RFC5246]
50	decode_error	Y	[RFC5246]
51	decrypt_error	Y	[RFC5246]
52-59	Unassigned		
60	export_restriction_RESERVED	Y	[RFC5246]
61-69	Unassigned		
70	protocol_version	Y	[RFC5246]
71	insufficient_security	Y	[RFC5246]
72-79	Unassigned		
80	internal_error	Y	[RFC5246]
81-85	Unassigned		
86	inappropriate_fallback	Y	[RFC7507]
87-89	Unassigned		
90	user_canceled	Y	[RFC5246]
91-99	Unassigned		
100	no_renegotiation	Y	[RFC5246]
101-109	Unassigned		
110	unsupported_extension	Y	[RFC5246]
111	certificate_unobtainable	Y	[RFC6066]
112	unrecognized_name	Y	[RFC6066]
113	bad_certificate_status_response	Y	[RFC6066]
114	bad_certificate_hash_value	Y	[RFC6066]
115	unknown_psk_identity	Y	[RFC4279]
116-255	Unassigned		

Issue 1138/PR 1145: FrozenArray, sequence and SameObject (foolip)

- This should be merged after confirming that none of the renamed things have been implemented anywhere.

```
1356 1356      readonly      attribute RTCIceConnectionState    iceConnectionState;
1357 1357      readonly      attribute RTCPeerConnectionState    connectionState;
1358 1358      readonly      attribute boolean?                    canTrickleIceCandidates;
1359      -      static readonly attribute FrozenArray<RTCIceServer> defaultIceServers;
1359      +      static sequence<RTCIceServer> getDefaultIceServers ();
1360 1360      RTCConfiguration      getConfiguration ();
1361 1361      void                  setConfiguration (RTCCConfiguration configuration);
1362 1362      void                  close ();

4569 4572      <pre class="idl">[Serializable]
4570 4573      interface RTCCertificate {
4571 4574      readonly      attribute DOMTimeStamp expires;
4572      -      readonly      attribute FrozenArray<RTCDtlsFingerprint> fingerprints;
4575      +      sequence<RTCDtlsFingerprint> getFingerprints ();
4573 4576      // At risk due to lack of implementers' interest.
4574 4577      AlgorithmIdentifier getAlgorithm ();
4575 4578      };</pre>
```

WebRTC PC Issues

- [Issue 1128](#): Does MSID still work? (Taylor)

Issue 1128: Does MSID still work? (Taylor)

- Sometimes remote track IDs are generated, because:
 - The RtpReceiver's track is created at construction time, for “early media” use cases.
 - Its ID can't change after it's created.
- This means that applications can't reliably use track IDs to identify tracks between two endpoints. They must use MIDs or m= section indices instead.
- Stream IDs are still ok, because the remote stream is only created by setting a remote description.

Issue 1128: Does MSID still work? (Taylor)

- What should we do about this?
 - Leave things as they are, and document this behavior better so it doesn't keep surprising people?
 - Get rid of track ID signaling altogether?
 - Try to fix the problem in some way (for example, only create the track/generate the ID when the “track” attribute getter is called)?

Media Capture Pull Requests

- [Issue 444/PR 445](#): Add autoGainControl & noiseSuppression constraints (jib)

Issue 444/PR 445: autoGainControl & noiseSuppression (jib)

- ```
navigator.mediaDevices.getUserMedia({ audio: {
 - googNoiseSuppression: false,
 - googAutoGainControl: false,
 - mozNoiseSuppression: false,
 - mozAutoGainControl: false,
 + noiseSuppression: false,
 + autoGainControl: false,
 }
});
```
- Two mature independent implementations. Avoid vendor-prefix in .getSupportedConstraints()
- Two most common mic filters, great for WebRTC speech, terrible for music & anything else.
- Improved web control over browser default behaviors. Use-case: [remote stethoscope](#).
- Enables high fidelity recording control for e.g. MediaRecorder, Web Audio & peer connection.
- [Firefox headphone demo](#) (with .getSettings()/applyConstraints())

# Thank you

Special thanks to:

Harald!!!

W3C/MIT for WebEx

WG Participants, Editors & Chairs