

VP9 Is Now Available in WebRTC



By Sam Dutton

Sam is a Developer Advocate

Two years ago Chrome enabled support for the VP9 codec. From Chrome 48 on desktop and Android, VP9 will be an optional video codec for video calls using WebRTC.

While VP9 uses the same basic blueprint as previous codecs, the WebM team has packed improvements into VP9 to get more quality out of each byte of video. For instance, the encoder prioritizes the sharpest image features, and the codec now uses asymmetric transforms to help keep even the most challenging scenes looking crisp and block-free.

With VP9, internet connections that are currently able to serve 720p without packet loss or delay will be able to support a 1080p video call at the same bandwidth. VP9 can also reduce data usage for users with poor connections or expensive data plans, requiring in best cases only 40% of the bitrate of VP8.

You can see how VP8 calls compare with VP9 in the screenshot below of recordings we made with the WebRTC encoder settings, showing 30% bitrate savings:

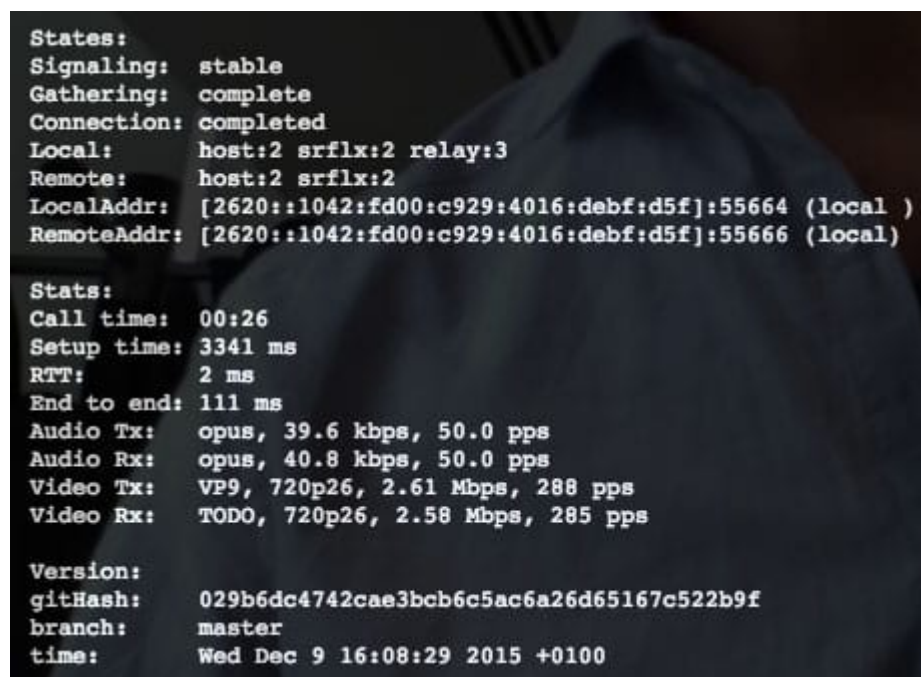


The codec for a WebRTC call, along with other media settings such as bitrate, is negotiated between caller and callee by exchanging Session Description Protocol (SDP) metadata messages that describe the media capabilities of the client.

This handshaking process — exchanging media capabilities — is known as offer/answer. For example, a caller might send an offer (an SDP message) stating a preference for VP9, with VP8 as a fallback. If the answer confirms that the callee can handle VP9, the video call can proceed using VP9. If the callee responds with an answer that it can only use VP8, the call will proceed with VP8.

To see this in action, take a look at the code for the canonical WebRTC video chat application [appr.tc](#).

Top tip: In [appr.tc](#), you can press **I** to get information about call state including signaling and codec details:



```
States:
Signaling:  stable
Gathering:  complete
Connection: completed
Local:      host:2 srflx:2 relay:3
Remote:     host:2 srflx:2
LocalAddr:  [2620::1042:fd00:c929:4016:debf:d5f]:55664 (local )
RemoteAddr: [2620::1042:fd00:c929:4016:debf:d5f]:55666 (local)

Stats:
Call time:  00:26
Setup time: 3341 ms
RTT:        2 ms
End to end: 111 ms
Audio Tx:   opus, 39.6 kbps, 50.0 pps
Audio Rx:   opus, 40.8 kbps, 50.0 pps
Video Tx:   VP9, 720p26, 2.61 Mbps, 288 pps
Video Rx:   TODO, 720p26, 2.58 Mbps, 285 pps

Version:
gitHash:    029b6dc4742cae3bcb6c5ac6a26d65167c522b9f
branch:     master
time:       Wed Dec 9 16:08:29 2015 +0100
```

In [appcontroller.js](#), VP9 is set as the preferred codec unless a `vsc` or `vrc` parameter is specified in the URL:

```
AppController.prototype.loadUrlParams_ = function() {
  // ...
  var DEFAULT_VIDEO_CODEC = 'VP9';
  // ...
  this.loadingParams_.videoSendCodec = urlParams['vsc'];
  // ...
  this.loadingParams_.videoRecvCodec = urlParams['vrc'] || DEFAULT_VIDEO_CODEC;
}
```



In [sdputils.js](#) the custom codec value (if specified) is then used for the SDP metadata:

```
function maybePreferVideoSendCodec(sdp, params) {  
  return maybePreferCodec(sdp, 'video', 'send', params.videoSendCodec);  
}  
  
function maybePreferVideoReceiveCodec(sdp, params) {  
  return maybePreferCodec(sdp, 'video', 'receive', params.videoRecvCodec);  
}
```

The [maybePreferCodec\(\)](#) function used here sets values for the requested codec in the text of the SDP metadata. SDP is verbose and not designed to be human readable, but you can view the SDP used by [appr.tc](#) from the DevTools console once a call has been made. The important part with regard to codecs is the *m line*:

```
{  
  "sdp": "v=0\r\no=- 9188830394109743399 2 IN IP4 127.0.0.1\r\ns ... m=video ..."  
  "type": "offer"  
}
```

Using [appr.tc](#) with its default settings in a recent version of Chrome, you will see that VP9 is the first codec listed in the SDP *m line* — followed by VP8, which Chrome can also use. If you set VP8 as the preferred codec (via URL parameters in [appr.tc](#), for example) VP8 will be listed first instead.

Find out more

- [The WebM Project](#): VP9 resources
- [webrtc.org](#): home of the WebRTC project
- [g.co/webrtc](#): links to other WebRTC resources
- [webrtc.github.io/samples](#): WebRTC samples live
- [test.webrtc.org](#): open source device and network testing tools and bug reporting framework

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](#), and code samples are licensed under the [Apache 2.0 License](#). For details, see our [Site Policies](#). Java is a registered trademark of Oracle and/or its affiliates.

Last updated July 2, 2018.