W3C WebRTC WG Meeting

February 20, 2024
8 AM - 10 AM

Chairs: Bernard Aboba
Harald Alvestrand
Jan-Ivar Bruaroey

W3C WG IPR Policy

- This group abides by the W3C Patent Policy https://www.w3.org/Consortium/Patent-Policy/
- 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 February 2024 interim meeting of the W3C WebRTC WG, at which we will cover:
    - WebRTC-PC, Screen-Share, Media-Recorder, Mediacapture-Main
- Future meetings:
    - March 26
    - April 23

About this Virtual Meeting

- Meeting info:
  - https://www.w3.org/2011/04/webrtc/wiki/February_20_2024
- Link to latest drafts:
  - https://w3c.github.io/mediacapture-main/
  - https://w3c.github.io/mediacapture-extensions/
  - https://w3c.github.io/mediacapture-image/
  - https://w3c.github.io/mediacapture-output/
  - https://w3c.github.io/mediacapture-screen-share/
  - https://w3c.github.io/mediacapture-record/
  - https://w3c.github.io/webrtc-pc/
  - https://w3c.github.io/webrtc-extensions/
  - https://w3c.github.io/webrtc-stats/
  - https://w3c.github.io/mst-content-hint/
  - https://w3c.github.io/webrtc-priority/
  - https://w3c.github.io/webrtc-nv-use-cases/
  - https://github.com/w3c/webrtc-encoded-transform
  - https://github.com/w3c/mediacapture-transform
  - https://github.com/w3c/webrtc-svc
  - https://github.com/w3c/webrtc-ice
- Link to Slides has been published on WG wiki
- Scribe? IRC http://irc.w3.org/ Channel: #webrtc
- The meeting is (still) being recorded. The recording will be public.
- Volunteers for note taking?

W3C Code of Conduct

- This meeting operates under W3C Code of Ethics and Professional Conduct

- We're all passionate about improving WebRTC and the Web, but let's all keep the conversations cordial and professional

Virtual Interim Meeting Tips

This session is (still) being recorded

- Click Raise hand to get into the speaker queue.
- Click Lower hand to get out of the speaker queue.
- Please wait for microphone access to be granted before speaking.
- If you jump the speaker queue, you will be muted.
- Please use headphones when speaking to avoid echo.
- Please state your full name before speaking.
- Poll mechanism may be used to gauge the “sense of the room”.

Understanding Document Status

- Hosting within the W3C repo does ***not*** imply adoption by the WG.
  - WG adoption requires a Call for Adoption (CfA) on the mailing list.
- Editor's drafts do ***not*** represent WG consensus.
  - WG drafts ***do*** imply consensus, once they're confirmed by a Call for Consensus (CfC) on the mailing list.
  - Possible to merge PRs that may lack consensus, if a note is attached indicating controversy.

Issues for Discussion Today

- 08:10 - 08:50 AM WebRTC-PC: Codec Issues & PRs (Harald)
- 08:50 - 09:10 AM Screen-Share & Screen-Share Extensions (Jan-Ivar)
- 09:10 - 09:30 AM MediaStream Recording (Jan-Ivar)
- 09:30 - 09:50 AM Mediacapture-Main (Jan-Ivar)
- 09:50 - 10:00 AM Wrapup and Next Steps (Chairs)

Time control:

- A warning will be given 2 minutes before time is up.
- Once time has elapsed we will move on to the next item.

WebRTC-PC

Start Time: 08:10 AM

End Time: 08:50 AM

For Discussion Today

- IETF (informative)
    - Issue 22: Issues with receive-only codecs (Bernard)
- WebRTC-PC
    - Issue 2925: Modify the codec description model to ease describing changes (Harald)
    - Issue 2929: Should media capabilities influence what is exposed in what is exposed in WebRTC offers and answers (Youennf)
    - Issue 2933: Existing setCodecPreferences note is wrong (Philipp)
    - Issue 2939: Proposing setCodecPreferences to deal with both send and recv codecs (Philipp)

Issue 22: Issues with receive-only codecs (Bernard)

- AVTCORE WG discussion relating to JSEPbis Section 4.2.6:

```
Note that setCodecPreferences does not directly affect which codec the
implementation decides to send. It only affects which codecs the implementation
indicates that it prefers to receive, via the offer or answer.
```

- Thread about whether this is correct (**for send-only m-line, consensus trending toward "no"**).
- Discussion of whether SCP() description in WebRTC-PC is mostly correct sans Issues (consensus trending toward "yes")
- Next step proposed at AVTCORE WG meeting: Discuss with RTCWEB WG whether change to JSEPbis is needed.

Issue 2925: Modify the codec description model to ease describing changes (Harald)

- The description of codec negotiation is rather informal
- When dealing with sendonly codecs especially, we’ve found major issues in describing sensible behavior
- PR 2935 attempts to fix that
  - Define a conceptual list of “all the configurations we support” for sending and receiving - owned by Sender and Receiver, respectively
  - What to enable is implementation dependent
  - Those configurations we are asked for get enabled
- Defines negotiation more precisely without being codec specific

Issue 2925: Modify the codec description model to ease describing changes (continued)

Things that should be easier with the new description:

- Enabling codecs that are not enabled-by-default
  - Remote offers can enable codecs (done today)
  - We can use media capabilities to get codecs (see m-c/#186)
  - We can add new API to enable codecs (or modify setSendCodec / setCodecPreferences)
- Adding user defined codecs from transforms or send APIs
  - Conceptually, we add them to the sender/receiver lists of codecs
- Keeping the list of codecs in SDP down to size
  - Can make the default smaller when we can enable more
  - Can add api to disable a codec configuration

Just the clearer exposition should ease testing conformance and interoperability

PR 186 in encoded-transform: User defined codecs

- A transform can have a list of input and output codecs attached
- When attached to a sender, add the output codecs to its SendCodecs list and enable them
- When attached to a receiver, add the input codecs to its ReceiveCodecs list and enable them
- If not an exact match for an existing codec, only those transceivers will ever see those packets
- All other machinery works as before

Described in “sdp-explainer.md” file of PR

OK to create spec change and merge?

Issue 2929: Should media capabilities influence what is exposed in what is exposed in WebRTC offers and answers (Youenn)

- Privacy principle: data minimization
  - https://www.w3.org/TR/privacy-principles/#data-minimization

- The whole list of A/V codecs is exposed at once using WebRTC
  - Via getCapabilities and/or SDP
  - More and more codecs (AV.1, HEVC)

- Media playback approach
  - Web site queries whether a configuration is supported via Media Capabilities
  - When web site finds a suitable supported configuration, it uses it
    - Web site can also try using each configuration directly

- Can WebRTC use Media Capabilities like is done for media playback?

Issue 2929: Should media capabilities influence what is exposed in what is exposed in WebRTC offers and answers (Youenn)

- Data minimization approach
  - A `fixed` list of exposed-by-default codecs for all user agents
    - This list should ideally not provide meaningful information about the user
  - Web site uses Media Capabilities to query for supported codecs
  - Web site enables explicitly codecs via a dedicated WebRTC API

- Advantages
  - Reduce the exposure of what gets exposed
  - Approach is probably fine for SFUs (limited set of supported configurations)
  - Changing default exposure of codecs may become practical

- Disadvantages
  - New codecs may no longer be used by default
  - UA-to-UA call negotiation may not yield to best setup without website's help

Issue 2933: Existing setCodecPreferences NOTE is wrong and should be deleted (Philipp)

- sCP note is rather vague
  - Due to a recommendation in [SDP], calls to createAnswer SHOULD use only the common subset of the codec preferences and the codecs that appear in the offer. For example, if codec preferences are "C, B, A", but only codecs "A, B" were offered, the answer should only contain codecs "B, A". However, [RFC8829] (section 5.3.1.) allows adding codecs that were not in the offer, so implementations can behave differently.
  - RFC 3264 / RFC 8829 allow answering with additional codecs
- Proposal: remove the note
  - This is already covered by JSEP
  - Codec negotiation clarified in issue 2925

Issue 2939: setCodecPreferences to deal with both send and recv codecs (Philipp)

- See also issues 2937 + 2938
- sCP influences receive preferences per JSEP
    - see PR 2926
- What happens if do sCP and then set the transceiver to sendonly?
    - Direction filtering may result in an empty set of codecs (due to sendonly/recvonly codecs). Then what?
    - Surprising for developers, JSEP clarification needed

Issue 2939: setCodecPreferences to deal with both send and recv codecs (Philipp)

- Avoid the situation by…
  - SCP and direction setting could sanity check and throw
  - createOffer could throw
  - if we have nothing to offer we reject the m= section (effective in SLD)
  - offer one of the MTI codecs as a "backup"
  - SCP could require that at least one codec is sendrecv, in which case there is no filtering that could remove it, and changing direction is always safe
- Any preferences?
- Discuss more in github issue?

Discussion (End Time: 08:50)

-

Screen-Capture (Jan-Ivar)

Start Time: 08:50 AM

End Time: 09:10 AM

Screen-capture Test Status

https://wpt.fyi/results/screen-capture

Path	Chrome 123 Linux 20.04 d643cf9 Feb 19, 2024	Edge 123 Windows 10.0 d643cf9 Feb 19, 2024	Firefox 124 Linux 20.04 d643cf9 Feb 19, 2024	Safari 188 preview macOS 13.6 d643cf9 Feb 19, 2024
capture-controller-event-target.https.window.html	3 / 3	3 / 3	0 / 1	0 / 1
delegate-request.https.sub.html	0 / 3	0 / 3	2 / 3	2 / 3
getdisplaymedia-after-discard.https.html	1 / 1	1 / 1	1 / 1	1 / 1
getdisplaymedia-capture-controller.https.window.html	51 / 51	51 / 51	0 / 51	0 / 51
getdisplaymedia-framerate.https.html	1 / 1	1 / 1	1 / 1	0 / 1
getdisplaymedia.https.html	63 / 64	63 / 64	49 / 64	44 / 64
historical.https.html	1 / 1	1 / 1	1 / 1	1 / 1
idlharness.https.window.html	23 / 23	23 / 23	16 / 23	16 / 23
permissions-policy-audio.https.sub.html	5 / 5	5 / 5	5 / 5	5 / 5
permissions-policy-audio+video.https.sub.html	5 / 5	5 / 5	5 / 5	5 / 5
permissions-policy-video.https.sub.html	5 / 5	5 / 5	5 / 5	5 / 5
Subtest Total	158 / 162	158 / 162	85 / 160	79 / 160

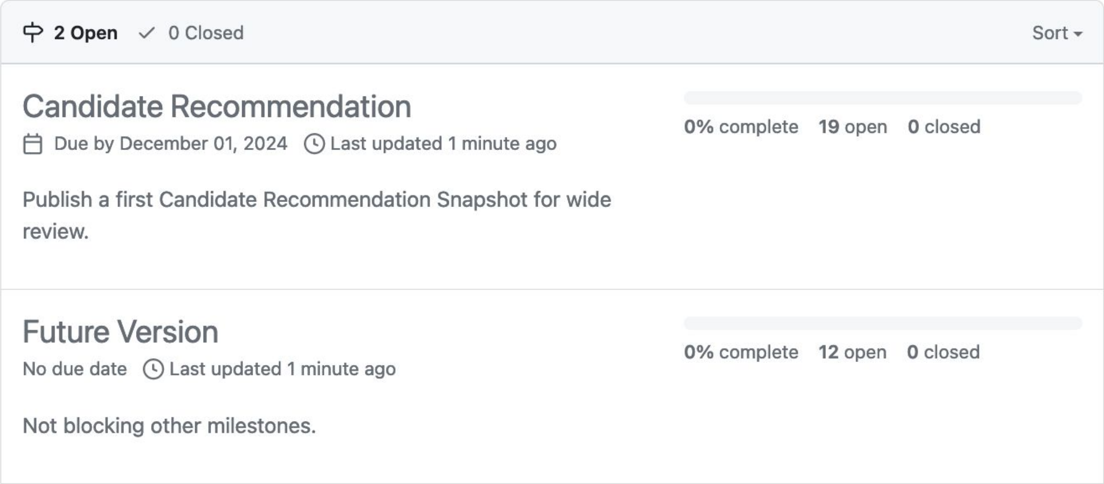
For Discussion Today

- Screen-capture (aka Mediacapture-screen-share)
  - Issue triage and milestones
  - Issue 297: Should we have a screenshare extension spec?
  - Issue 281: Distinguish cancellations from absent OS permissions

Issue triage and milestones (Screen-capture)

Screen Capture is still a Working Draft, despite its maturity in implementations. But is *active*.

Screen-capture has 31 open issues. To assist further triage, two milestones have been added:



19 issues are identified as blocking Candidate Recommendation.
12 issues earlier identified by chairs as enhancements are identified as not blocking CR.

(Chairs can sort issues in milestones. This is an organizing tool. Feel free to challenge.)

Issue 297: Should we have a screenshare extension spec?

Chairs would like to expedite this document to Candidate Recommendation (and eventual Recommendation), by splitting out enhancement requests into a follow-up.

In the past we've used extension repos for this (even pre-REC):

- webrtc-pc → webrtc-extensions
- mediacapture-main → mediacapture-extensions

Should we do the same here, and transfer the (12) enhancements?

If so, (Bikeshed!) do we go by github name or TR name?

- mediacapture-screen-share → mediacapture-screen-share-extensions ?
- screen-capture → screen-capture-extensions ?

Open to alternatives.

Issue 281: Distinguish cancellations from absent OS permissions

Applications want to distinguishing missing OS permissions from user cancellation.

Firefox rejects with `NotFoundError` in this situation, based on: *"If no sources of type T are available [to the UA (interpretation)]"*. Same with camera and microphone.

OS permissions may be out of reach on some systems (due to admins or parents). Distinct

For screen-sharing, this should be unambiguous in normal cases, and for camera/mic sites can disambiguate using enumerateDevices() which appears to work wo/OS permission.

Makes sense? If so, the proposal is to add a note about this.

Discussion (End Time: 09:10)

-

MediaStream-Recording (Jan-Ivar)

Start Time: 09:10 AM

End Time: 09:30 AM

MediaStream Recording Test Status

https://wpt.fyi/results/mediacapture-record

Path	Chrome 123 Linux 20.04 d643cf9 Feb 19, 2024	Edge 123 Windows 10.0 d643cf9 Feb 19, 2024	Firefox 124 Linux 20.04 d643cf9 Feb 19, 2024	Safari 188 preview macOS 13.6 d643cf9 Feb 19, 2024
BlobEvent-constructor.html	3 / 3	3 / 3	3 / 3	3 / 3
idlharness.window.html	60 / 60	60 / 60	56 / 60	58 / 60
MediaRecorder-bitrate.https.html	18 / 18	18 / 18	18 / 18	18 / 18
MediaRecorder-canvas-media-source.https.html?mimeType=''	0 / 1	0 / 1	0 / 1	0 / 1
MediaRecorder-canvas-media-source.https.html?mimeType=video/mp4;codecs=avc1,mp4a.40.2	0 / 1	0 / 1	0 / 1	0 / 1 ⚠
MediaRecorder-canvas-media-source.https.html?mimeType=video/webm;codecs=av1,opus	1 / 1	1 / 1	0 / 1	0 / 1
MediaRecorder-canvas-media-source.https.html?mimeType=video/webm;codecs=vp8,opus	1 / 1	1 / 1	0 / 1 ⚠	0 / 1
MediaRecorder-canvas-media-source.https.html?mimeType=video/webm;codecs=vp9,opus	1 / 1	1 / 1	0 / 1	0 / 1
MediaRecorder-creation.https.html	3 / 3	3 / 3	3 / 3	3 / 3
MediaRecorder-destroy-script-execution.html	4 / 4	4 / 4	4 / 4	3 / 4
MediaRecorder-detached-context.html	1 / 1	1 / 1	0 / 1	0 / 1
MediaRecorder-disabled-tracks.https.html	3 / 3	3 / 3	3 / 3	3 / 3
MediaRecorder-error.html	4 / 4	4 / 4	1 / 4	1 / 4

MediaStream Recording Test Status (cont'd)

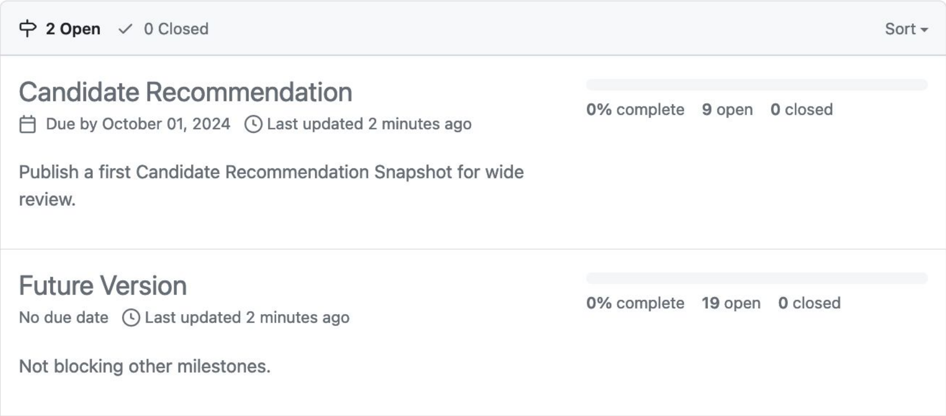
https://wpt.fyi/results/mediacapture-record

Path	Chrome 123 Linux 20.04 d643cf9 Feb 19, 2024	Edge 123 Windows 10.0 d643cf9 Feb 19, 2024	Firefox 124 Linux 20.04 d643cf9 Feb 19, 2024	Safari 188 preview macOS 13.6 d643cf9 Feb 19, 2024
MediaRecorder-disabled-tracks.https.html	3 / 3	3 / 3	3 / 3	3 / 3
MediaRecorder-error.html	4 / 4	4 / 4	1 / 4	1 / 4
MediaRecorder-events-and-exceptions.html?mimeType=''	0 / 1	0 / 1	0 / 1	0 / 1
MediaRecorder-events-and-exceptions.html?mimeType=video/mp4;codecs=avc1,mp4a.40.2	0 / 1	0 / 1	0 / 1	0 / 1
MediaRecorder-events-and-exceptions.html?mimeType=video/webm;codecs=av1,opus	1 / 1	1 / 1	0 / 1	0 / 1
MediaRecorder-events-and-exceptions.html?mimeType=video/webm;codecs=vp8,opus	1 / 1	1 / 1	0 / 1 ⚠	0 / 1
MediaRecorder-events-and-exceptions.html?mimeType=video/webm;codecs=vp9,opus	1 / 1	1 / 1	0 / 1	0 / 1
MediaRecorder-mimetype.html	66 / 66	66 / 66	66 / 66	63 / 66
MediaRecorder-pause-resume.html?mimeType=''	4 / 4	4 / 4	4 / 4	4 / 4
MediaRecorder-pause-resume.html?mimeType=video/mp4;codecs=avc1,mp4a.40.2	4 / 4	4 / 4	4 / 4	2 / 2
MediaRecorder-pause-resume.html?mimeType=video/webm;codecs=av1,opus	2 / 2	2 / 2	4 / 4	4 / 4
MediaRecorder-pause-resume.html?mimeType=video/webm;codecs=vp8,opus	2 / 2	2 / 2	0 / 2 ⚠	4 / 4
MediaRecorder-pause-resume.html?mimeType=video/webm;codecs=vp9,opus	2 / 2	2 / 2	4 / 4	4 / 4
MediaRecorder-peerconnection-no-sink.https.html	1 / 3	1 / 3	3 / 3	2 / 3 ⚠
MediaRecorder-peerconnection.https.html	16 / 16	16 / 16	14 / 14	1 / 11 !
MediaRecorder-start.html	1 / 1	1 / 1	1 / 1	0 / 1
MediaRecorder-stop.html?mimeType=''	0 / 8	0 / 8	0 / 8	0 / 8
MediaRecorder-stop.html?mimeType=video/mp4;codecs=avc1,mp4a.40.2	0 / 8	0 / 8	0 / 8	8 / 8
MediaRecorder-stop.html?mimeType=video/webm;codecs=av1,opus	8 / 8	8 / 8	0 / 8	0 / 8
MediaRecorder-stop.html?mimeType=video/webm;codecs=vp8,opus	8 / 8	8 / 8	0 / 8 ⚠	0 / 8
MediaRecorder-stop.html?mimeType=video/webm;codecs=vp9,opus	8 / 8	8 / 8	0 / 8	0 / 8
MediaRecorder-video-key-frame-distance.html	2 / 2	2 / 2	0 / 2	0 / 2
passthrough/	4 / 5	4 / 5	0 / 5	1 / 5
Subtest Total	230 / 253	230 / 253	188 / 255	182 / 252

Issue triage and milestones (MediaCapture-record)

MediaStream Recording remains a Working Draft, despite its maturity in implementations. But unlike Screen-Capture, there's no new interest, having been overtaken by WebCodecs.

Chairs would like to rush it to CR by documenting what exists and closing outstanding issues. MediaCapture-record has 28 open issues. To assist triage, two milestones have been added:



7 issues are identified as blocking CR.
19 issues are identified as not blocking CR (targeted for closing).

We'll try our best to minimize need for vendor assistance in closing out remaining CR issues.

For Discussion Today

- MediaStream-Recording (aka mediacapture-record
  - Issue 202: deprecate isTypeSupported

Discussion (End Time: 09:30)

-

MediaCapture-Main & misc (Jan-Ivar)

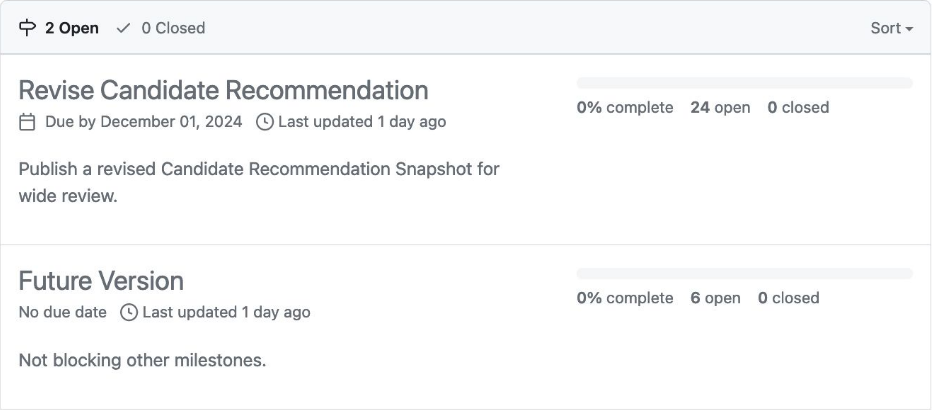
Start Time: 09:30 AM

End Time: 09:50 AM

Issue triage and milestones (MediaCapture-main)

MediaCapture Streams is on its 4th Candidate Recommendation, is mature in implementations, and already has a MediaCapture Extensions spillover repo. Defines not just camera and microphone, but the model for other specs, hence activity. Also devices are hard.

MediaCapture-main has 30 open issues. To assist triage, two milestones have been added:



24 issues are identified as blocking CR.
6 issues are identified as not blocking CR (mostly editorial).

Can we burn down issues and get to (last) revised CR and Proposed Recommendation?

For Discussion Today

- Mediacapture-from-element
  - Issue 65: captureStream on OffscreenCanvas
- Mediacapture-extensions
  - PR 26: Expose MediaStream in workers
- Mediacapture-main
  - Issue 974: How should MediaStreamTrack interact with BFCache?
  - PR 988: Add guidance for defining a new source of MediaStreamTrack

Issue 65: captureStream on OffscreenCanvas (Jan-Ivar)

This issue is tracked by TAG. We support this:

```
const canvas = document.createElement("canvas");
const stream = canvas.captureStream();
```

…so why not this?

```
const canvas = new OffscreenCanvas(640, 480);
const stream = canvas.captureStream();
```

Use case: Rendering to video in workers, or anywhere the canvas is a means to an end.

E.g. MediaStreamTrackProcessor can be shimmed using OffscreenCanvas today, but VideoTrackGenerator cannot (relies on canvas, hence stuck on main-thread, modulo MST)

PR 26: Expose MediaStream in workers

OffscreenCanvas is exposed in (Window,Worker), so for this to work in workers:

```
const canvas = new OffscreenCanvas(640, 480);
const stream = canvas.captureStream();
```

...MediaStream (which captureStream returns) would need to be exposed to workers.

We have a PR for this that has been waiting for a reason.

Transfer is NOT included in this PR, as transfer may not be necessary or a good idea. E.g. would tracks in the stream be transferred, even if shared with other streams? Workaround:

```
// transferred from a worker
worker.onmessage = ({data: {track}}) => video.srcObject = new MediaStream([track]);
```

Issue 974: How should MediaStreamTrack interact with BFCache?

Problem: Pages with live cam/mic/screen-share tracks reload ↻ on back/forward navigation

Proposal: End cam/mic/screen-share tracks but keep the page salvageable (BF-cacheable). Queue an ended event to fire if the page is ever restored (Safari already does this)

Web compat:

The ended event already exists and can fire for other reasons today, so applications *should* already be handling it, even though ***many are not:***

Does website recover from cam+mic `ended` ?	cam	mic
Meet (in room)	✅ automatic	❌ unrecoverable
Zoom (in room)	✅ after user toggle	❌ unrecoverable
Teams (in room)	✅ after user toggle	✅ after user toggle
Webex (in room)	✅ after user toggle	❌ unrecoverable
Whereby (in room)	❌ unrecoverable	✅ automatic
Jitsi (in room)	❌ unrecoverable[1]	❌ unrecoverable[1]
Facebook (calling...)	✅ after toggle in ⚙️	✅ after toggle in ⚙️

Of course ↻ is a fine workaround, but some outreach would seem necessary to improve this.

WebCompat seems less of a concern, given these websites are arguably broken today. BFCache will make the symptom worse, but likely improve the chances of websites fixing this. Win?

[PR 988](): Add guidance for defining a new source of MediaStreamTrack

1: After the application has invoked the `stop()` method on a `MediaStreamTrack` object, or once the source of a `MediaStreamTrack` ends production of live samples to its tracks, whichever is sooner, a `MediaStreamTrack` is said to be ***ended***.

For camera and microphone sources, the reasons for a source to end besides `stop()` are implementation-defined (e.g., because the user rescinds the permission for the page to use the local camera, or because the User Agent has instructed the track to end for any reason).

2: ***Muted*** refers to the input to the `MediaStreamTrack`. Live samples *MUST NOT* be made available to a `MediaStreamTrack` while it is muted.

`Muted` is outside the control of web applications, but can be observed by the application by reading the `muted` attribute and listening to the associated events `mute` and `unmute`. The reasons for a `MediaStreamTrack` to be muted are defined by its source.

For camera and microphone sources, the reasons to mute are implementation-defined. This allows user agents to implement privacy mitigations in situations like: the user pushing a physical mute button on the microphone, the user closing a laptop lid with an embedded camera, the user toggling a control in the operating system, the user clicking a mute button in the User Agent chrome, the User Agent (on behalf of the user) mutes, etc.

PR 988: Add guidance for defining a new source of MediaStreamTrack

> NOTE
>
> This does not apply to sources defined in other specifications. Other specifications need to define their own steps to set a track's muted state if desired.

3:

§ 17.4 Defining a new source of `MediaStreamTrack`

Other specs can define new sources of `MediaStreamTrack`. At a minimum, a new source of `MediaStreamTrack` will need to

- define a new API to create a MediaStreamTrack of the relevant `kind`s from this new source (`getUserMedia()` is dedicated to camera and microphone sources),
- declare which constrainable properties (see 4.3.8 Constrainable Properties), if any, are applicable to each `kind` of media this new source produces, and how they work with this source,
- describe how and when to set a track's muted state for this source,
- describe how and when to end tracks from this source,
- if capture of the source is a powerful feature requiring express permission, describe its permissions integration and permissions policy integration,
- if capture of the source poses a privacy concern, describe its privacy indicator requirements.

Discussion (End Time: 09:50)

-

Wrapup and Next Steps

Start Time: 09:50 AM

End Time: 10:00 AM

Next Steps

- Content goes here

Thank you

Special thanks to:

WG Participants, Editors & Chairs