

W3C WebRTC WG Meeting

May 20, 2025
8 AM - 10 AM

Chairs: Jan-Ivar Bruaroey
Youenn Fablet
Guido Urdaneta

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 May 2025 interim meeting of the W3C WebRTC WG, at which we will cover:
 -
- Future meetings:
 - [June 17 2025](#)
 - [July 15 2025](#)

About this Virtual Meeting

- Meeting info:
 - https://www.w3.org/2011/04/webrtc/wiki/May_20_2025
- 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 - 08:25 AM Transient activation after getDisplayMedia (Elad)
- 08:25 - 08:40 AM windowAudio (Johannes + Elad)
- 08:40 - 08:55 AM MediaCapture-main (Youenn)
- 08:55 - 09:20 AM MediaCapture-output (Youenn)
- 09:20 - 09:50 AM Grab Bag (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.

Transient activation after getDisplayMedia

Start Time: 08:10 AM

End Time: 08:25 AM

Reminder: Transient Activation

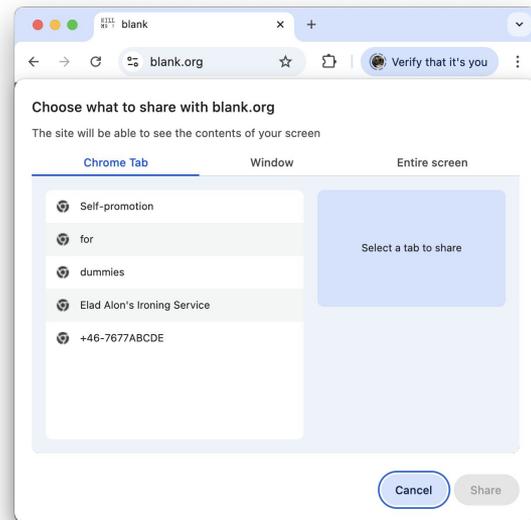
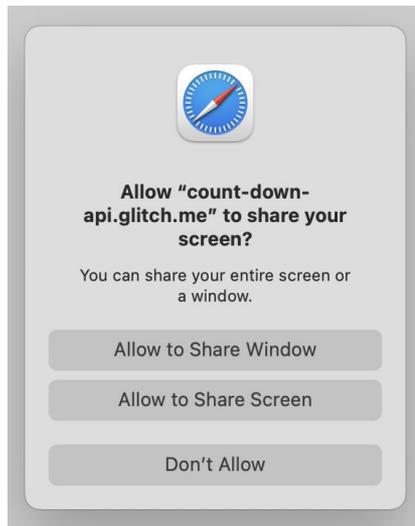
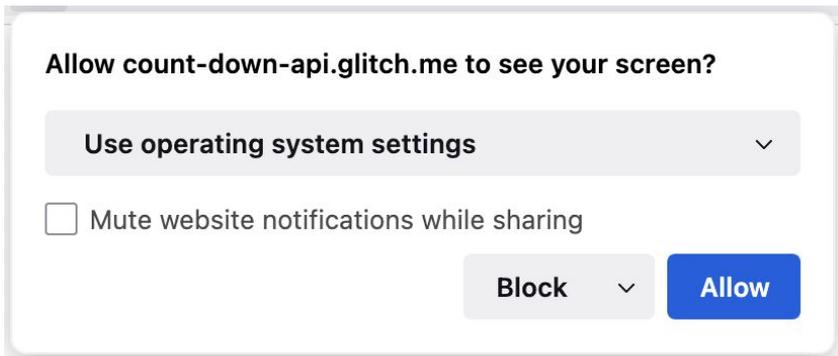
- MDN describes it as “user is currently interacting with the page.”
- When the user delivers a mouse-click or a keyboard key-press somewhere within the app, transient activation is conferred for a brief time.
 - How long depends on the implementation.
 - Chrome uses 5s atm.
- Web APIs are often gated on this requirement. Examples include:
 - [Fullscreen](#) (requestFullscreen)
 - [Document PiP](#) (requestWindow)
 - [Clipboard APIs](#)
 - Notably, [getDisplayMedia](#) itself.

Today's discussion

- **Not** up for discussion:
 - `getDisplayMedia` requires transient activation.
 - That's the status quo.
 - We are not seeking to change that.
- Up for discussion:
 - What happens immediately after `getDisplayMedia` resolves.

Observations

- When `getDisplayMedia` is called, the user is shown a UA prompt.
- The user can take arbitrarily long to interact with that prompt.
- The transient activation used to call `gDM`, even if not consumed, is liable to elapse while the user interacts with that prompt.



Room for Improvement

- When starting screen-sharing, a context-aware Web app can serve the user by taking additional actions.
 - Start playing media the user preconfigured to share.
 - Go full-screen if the user is recording a video game or a presentation.
 - Trigger PiP (Picture-in-Picture).
- Recall that if capturing another tab/window, the user might no longer be looking at the capturing Web app when capture starts.
- Ideally, the user should not need to...
 - Alt-tab back to the video-conferencing app to trigger PiP.
 - Record themselves interacting with the recording studio app and then later edit that interaction out.

Proposal

8. Queue a task on the user interaction task source to run the following steps:

1. If *controller* is not `null`, run the following steps:

1. Set *controller*.[[Source]] to *stream*'s video track's [[Source]].

2. Set *controller*.[[DisplaySurfaceType]] to the to *stream*'s video track's DisplayCaptureSurfaceType.

New: 2. Set the last activation timestamp to the current high resolution time.

3. Resolve *p* with *stream*.

Discussion (**End Time: 08:40**)

-

windowAudio (Johannes + Elad)

Start Time: 08:25 AM

End Time: 08:40 AM

getDisplayMedia(): Window audio hint

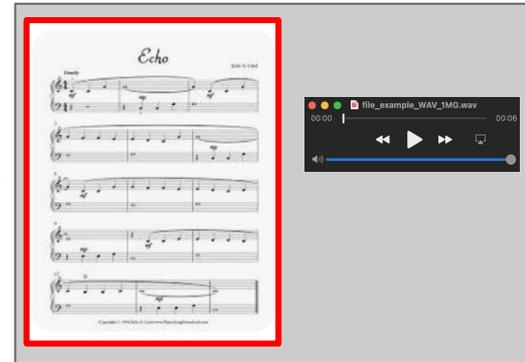
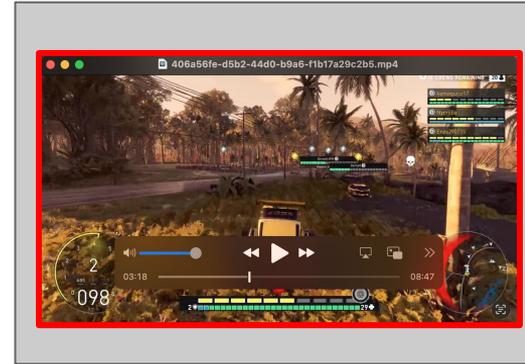
- `getDisplayMedia()` is an API to capture a user's display. Could be either a tab, window or monitor.
- Application may offer audio capture by setting an option `{audio: true}`. Could be either tab, system, or application audio.
- Some applications prefer to not capture system audio.

`SystemAudioPreferenceEnum` was added with

include	The application prefers that options to share system audio be offered to the user for monitor display surfaces.
exclude	The application prefers that options to share system audio not be offered to the user.

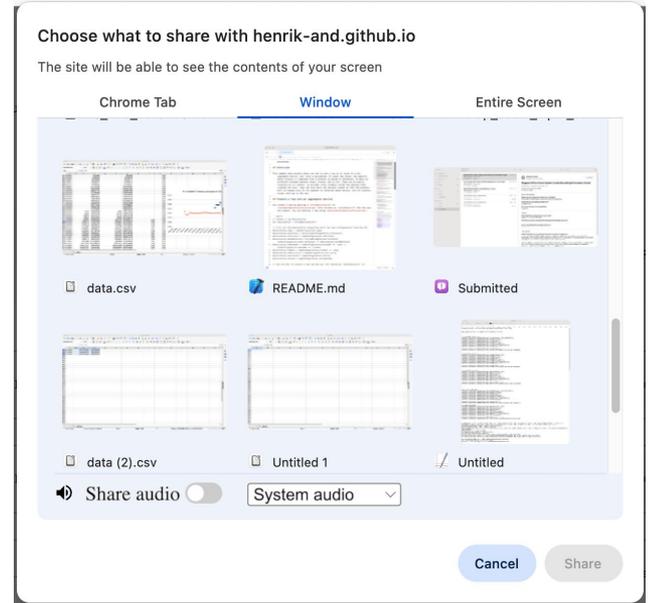
Window audio

- **Application** or **System** audio?
- Examples:
 - If a video player is captured, likely the user only intended to share a clip with sound.
=> **Application** audio.
 - If a user is recording themselves giving a lecture about music, they might be showing a PDF with sheet music while playing music through a music player.
=> **System** audio.



Window audio, continued

- Core principle is that the user decides.
- But, the user is often busy, so user agents aim to offer choices with reasonable defaults.
- The Web application often has more context about what the user is doing and can give a hint to the user agent.



Mock of window audio setting.

Proposal

- Add `WindowAudioPreferenceEnum` `windowAudio` to `DisplayMediaStreamOptions` where:

```
enum WindowAudioPreferenceEnum {  
    "application",  
    "system",  
    "exclude"  
};
```

application	The application prefers that the option to share application audio is offered to the user for window display surfaces.
system	The application prefers that the option to share system audio is offered to the user for window display surfaces.
exclude	The application prefers that no options to share audio be offered to the user for window display surfaces.

Discussion (**End Time: 08:40**)

-

MediaCapture-main

Start Time: 08:40 AM

End Time: 08:55 AM

Should a muted video track be allowed to deliver black frames to its sinks?

No interoperability for enabled=false video tracks

- <https://jsfiddle.net/6gkvsfc7>
- Chrome and Safari
 - HTMLVideoElement is receiving video tracks but is rendering black
 - Rvfc is called
 - HTMLVideoElement gets resized according track video frame size
- Firefox
 - HTMLVideoElement is not receiving video tracks
 - Rvfc is not called
 - HTMLVideoElement size remains as is until track is no longer muted/disabled

Should a muted video track be allowed to deliver black frames to its sinks?

Proposal: align with Firefox behavior & clarify the specification

- Sinks do not receive any video frame
- Sinks are expected to react to the signals themselves
 - HTMLVideoElement is rendering black but no rvfc/no size change
 - To be clarified in mediacapture-main
 - RTCRtpSender sends one black frame per second
 - Already covered in webrtc-pc
 - MediaStreamTrackProcessor
 - Nothing to do
 - MediaRecorder?
 - Which frame rate to use

Discussion (**End Time: 08:55**)

-

MediaCapture-output

Start Time: 08:55 AM

End Time: 09:20 AM

Setting tab-wide audio output

- Current issues
 - <https://github.com/w3c/mediacapture-output/issues/141>
 - <https://github.com/w3c/audio-session/issues/6>
- Most web applications want to target a single speaker
 - Convenient to do the switch within a single call
 - Better align with iOS & Android
- API sets the default speaker
 - `HTMLMediaElement.setSinkId` can override this default speaker

Setting tab-wide audio output

- Default speaker scope?
 - All media elements of the tab
 - Third-party iframes could change the default speaker of its top-level context
 - All media elements of the context and its children
- Current proposal
 - Only allow this API on the top level context until we reach consensus on the default speaker scope
- How to make progress?
 - And where: Media WG or WebRTC WG?

Discussion (**End Time: 09:20**)

-

Grab Bag

Start Time: 09:20 AM

End Time: 09:50 AM

Grab bag for discussion today

- webrtc-pc
 - [Issue 3049](#): Validate an ICE server url is missing length check for username
 - [Issue 3045](#): Inconsistent initialization of `transceiver.receiver.track.getSettings()`
- mediacapture-main
 - [Issue 1041](#): Clarify `resizeMode`: `none` \subseteq `crop-and-scale`

[Issue 3049](#) / [PR 3050](#): Validate an ICE server url is missing length check for username

PR: 7. If *parsedURL*'s `scheme` is "turn" or "turns", and either of *server.username* or *server.credential* are missing or their UTF-8 representations fail to conform to [RFC8489] section 14.3 and [RFC8265] section 4.1 respectively, then throw an `InvalidAccessError`.

[RFC 8489](#): "USERNAME is a variable-length value ... MUST contain a UTF-8-encoded [RFC3629] sequence of fewer than 509 bytes and MUST have been processed using the OpaqueString profile [RFC8265]."

[RFC 8265](#): "A password MUST NOT be zero bytes in length."

Means:

- `{username: chars509, credential: "foo"}` // `InvalidAccessError`
- `{username: "", credential: "foo"}` // `is valid`
- `{username: "", credential: ""}` // `InvalidAccessError`

Issue 3045: Inconsistent initialization of `transceiver.receiver.track.getSettings()`

The Constraining Pattern [disallows](#) uninitialized values: “A `[[Settings]]` internal slot, initialized to a `Settings` dictionary describing the currently active settings values for each constrainable property exposed, as explained under [Settings](#), or an empty dictionary if it has none” [exposed]

MediaCapture-main gives this guidance:

Proposal: Apply the same to `webrtc-pc`:

NOTE

It is recommended to look at existing implementations to select meaningful default values. Note that default values may differ based on the system, for instance desktop vs. mobile. At time of writing, [User Agent](#) implementations tend to use the following default values, which were chosen for their suitability for using `RTCPeerConnection` as a sink:

1. `width` set to 640.
2. `height` set to 480.
3. `frameRate` set to 30.
4. `echoCancellation` set to `true`.

```
> RTCPeerConnection().addTransceiver("video").receiver.track.getSettings()  
{ aspectRatio: 1.3333333333333333, frameRate: 30, height: 480, width: 640 }
```

Issue 1041 / PR 1042: Clarify resizeMode: none \subseteq crop-and-scale

A camera with **one** native capability of 640x480@30 and the constraint `{resizeMode: {exact: "crop-and-scale"}}` produces what? We want:

640x480@30 **not** 639x480@30

Intent: "none" is a filter. IOW,
native modes \subseteq crop-and-scale

Fix the input to SelectSettings
to reflect this. PR: 

3. Invoke and return the result of the [applyConstraints template method](#) where:

- In the [SelectSettings](#) algorithm,
 - *object* is the [MediaStreamTrack](#) on which this method was called, and
 - [settings dictionary](#) refers to a possible instance of the [MediaTrackSettings](#) dictionary. The [User Agent](#) **MUST NOT** include inherent unchangeable device properties as members unless they are in the [list of inherent constrainable track properties](#), or otherwise include device properties that **must not be exposed**.

NOTE

Other specifications may define constrainable properties that at times [must not be exposed](#).

- For every [settings dictionary](#) with [resizeMode](#) set to "none", the [User Agent](#) **MUST** include another otherwise identical [settings dictionary](#) with [resizeMode](#) set to "crop-and-scale". Constraining around non-native modes is not possible.

Discussion (**End Time: 09:50**)

-

Wrapup and Next Steps

Start Time: 09:50 AM

End Time: 10:00 AM

Thank you

Special thanks to:

WG Participants, Editors & Chairs