

W3C Media & Entertainment Interest Group Meeting

7 January 2025



Agenda

Date and time

7 January 2025, 15:00-16:00 UTC

IRC

<https://irc.w3.org/?channels=#me>

IRC Guide

<https://www.w3.org/wiki/IRC>

Code of Conduct

<https://www.w3.org/policies/code-of-conduct/>

Agenda

- Welcome
- Sync on the Web Community Group
- SVTA MSE issues review and next steps
- AOB

Sync on the Web Community Group

- TPAC 2024 Breakout hosted by Kensaku Komatsu:
 - <https://www.w3.org/2024/09/25-sync-on-web-minutes.html>
 - <https://www.w3.org/2024/Talks/TPAC/breakouts/sync-on-web.pdf>
- Web MIDI API and other input devices, e.g., Gamepad API. Latency, synchronization with media, including Media over QUIC. Real-time utilization of data from locally connected devices is extending beyond local clients to reach remote clients across the network. These protocol advancements enable the web to deliver orchestrated, immersive experiences that seamlessly integrate virtual and real-world elements in real time. To fully leverage these opportunities and ensure precise reproduction of such experiences on client devices, our current focus is on timing synchronization at the sender's side, particularly synchronizing data from locally connected devices.
- Community Group: <https://www.w3.org/community/sync-on-the-web/>

MSE

- Minutes from December meeting: <https://www.w3.org/2024/12/10-me-minutes.html>
- Topics:
 - Buffer capacity
 - Box parsing
 - Codec information
 - Dynamic addition/removal of Source Buffers
 - Multiple Source Buffers

Buffer capacity

- Related MSE "introspection API" feature requests:
- [#35](#) Report buffer changes in update events
- [#40](#) Needs event to notify when sourceBuffer needs more data
- [#172](#) Consider adding API for app to know how much room is left in the SourceBuffer
- [#259](#) Consider adding support for apps to get metadata about what is currently buffered (codec information)
- [MSE SourceBuffer Introspection Proposal](#)
- Managed MediaSource bufferedchange events
- Implementations that require appending full mdat box structures

Buffer Capacity

W3C Media & Entertainment IG Meeting

9

• Description:

- Media players maintain a forward and a backward buffer. The size of the buffer is usually configurable
- There is no API to query the maximum buffer for a platform
- A `QuotaExceededError` is thrown by the MSE when the buffer limit exceeded.

• Implications:

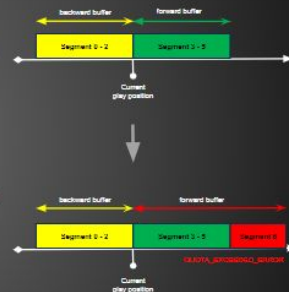
- Implementations need to try and error. No delayed scheduling possible.
- In the worst case, the downloaded data is discarded
- Buffer targets are dynamically adjusted once an error occurs

• Suggested Improvements:

- An API to query the maximum buffer size

• Related use cases:

- General buffer management in every MSE based player



Box parsing

- MSE [#189](#) Add Support for Media-Encoded Events
- [Media Timed Events TF](#) (2018-22)
- [Requirements for Media Timed Events](#)
- DataCue API [proposal](#) and [draft spec](#)
- Can we quantify the performance of JS or WASM box parsing?
- Subtitles and captions in MSE, 608/708 caption data. TPAC2024 [discussion](#)
- Related: WebCodecs [#24](#) API for containers? MEIG [#108](#) Media Containers API
- WebCodecs + Canvas based player
 - [#41](#) Support for content protection
- Suggest focusing on use-case level requirements, may lead to other proposed solutions

Box parsing

W3C Media & Entertainment IG Meeting

10

● Description:

- Some ISOBMFF boxes are interesting for the media player or the application
- As of today, media players implement their own box parser to extract relevant information from the ISOBMFF boxes

● Implications:

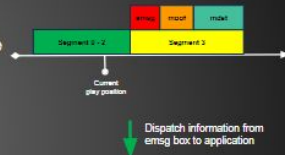
- Various Box parsers that implement the same functionality
- JavaScript based parsing can slow down client (WebAssembly is an option). Especially relevant in the context of low-latency streaming and chunked-transfer coding (moof,mdat parsing)

● Suggested Improvements:

- An API to register for specific ISOBMFF boxes and get a notification with the payload of the box

● Related use cases:

- EMSG metadata, PRFT latency adjustment, ELST handling, MOOV for timescale corrections



Codec information

- Related MSE "introspection API" feature request:

- [#259](#) Consider adding support for apps to get metadata about what is currently buffered

Codec information

W3C Media & Entertainment IG Meeting

11

- **Description:**

- A SourceBuffer is initialized with a specific codec string
- To keep existing SourceBuffers once the codec changes, a media player can use the `changeType()` method

- **Implications:**

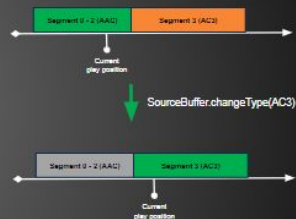
- Media players need to keep track of the current SourceBuffer codec and only apply `changeType()` if the new codec is not compatible. For instance, a switch from avc1 to avc3 does not require a call to `changeType()`

- **Suggested Improvements:**

- An API/attribute to query the current codec string and mime type of a SourceBuffer

- **Related use cases:**

- General SourceBuffer management in every media player
- Multiperiod DASH playback



Dynamic addition/removal of Source Buffers

Dynamic addition/removal of Source Buffers

12

W3C Media & Entertainment IG Meeting

- **Description:**

- Buffers are initialized at the beginning of the session
- After initialization, buffers can not be reliably removed / added

- **Implications:**

- A Player implementation must work with an existing setup of buffers or fully restart to change
- Fully restarting has an impact on the user for a simple case such as turning off audio
- There are workaround by carefully constructing "silent" or "black frame" payload

- **Suggested Improvements:**

- Allow to dynamically remove and (re-)add buffers

- **Related use cases:**

- Fully disable individual tracks, for example for background audio only playback or silent video only playback
- Support for text buffers (WIP?) might complicate this further

- [#160](#) Support playback through unbuffered ranges
-

Multiple Source Buffers

Multiple Source Buffers

W3C Media & Entertainment IG Meeting

13

- **Description:**

- Only one video / audio buffer can be attached and used
- Modern workflows

- **Implications:**

- Not perfectly conditioned content can not be appended immediately to a buffer
- A virtual buffer can be maintained on a client, but appended segments there will not be parsed
- Timing when to push from the virtual buffer to the main buffer can be tricky
- Gapless timeline consistency can be tricky

- **Suggested Improvements:**

- Allow to maintain multiple detached buffers on a media element and permit switching

- **Related use cases:**

- Server Guided Ad Insertion with not perfectly conditioned content
- Playlist support for media with overlapping timelines



- [#357](#) Detachable MediaSource proposal

Thank you!