Common Media Library (CML)

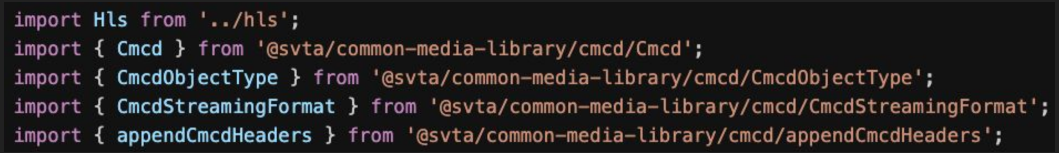
OSMART Workshop #2

Casey Occhialini

Principal Software Engineer @ Paramount

Common Media Library - Overview

The Common Media Library (CML) is a modern Javascript utility library for media playback.

- Written in TypeScript and modern build tools
- Compiled to ES6 with Typescript declarations
  - No bundling or transpiling.
- Focused on modularity. Only import what is needed.
  - 

```
import Hls from '../hls';
import { Cmcd } from '@svta/common-media-library/cmcd/Cmcd';
import { CmcdObjectType } from '@svta/common-media-library/cmcd/CmcdObjectType';
import { CmcdStreamingFormat } from '@svta/common-media-library/cmcd/CmcdStreamingFormat';
import { appendCmcdHeaders } from '@svta/common-media-library/cmcd/appendCmcdHeaders';
```

- API documentation derived from TSDoc
- Maintained by the SVTA, but open to everyone for contributions

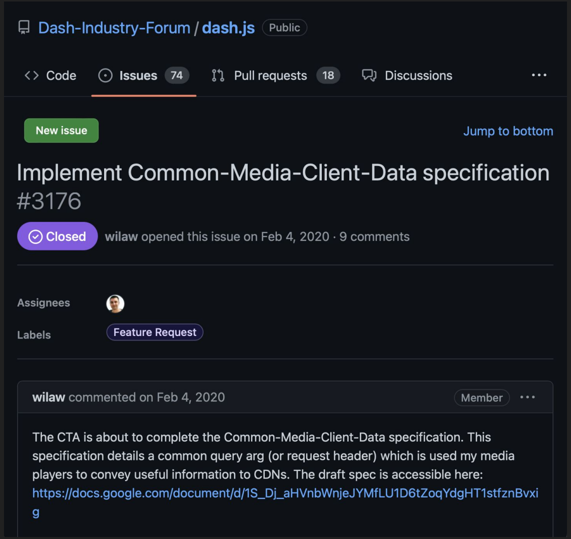
Reasoning

Reduce duplicate code across popular web players and create a single place for standards groups to request and oversee feature implementation

- Major open source projects like hls.js, dash.js, shaka-player, and video.js all support standard based features like ID3 parsing, CMCD, CMSD, CTA-608/708, etc.
- These common pieces of functionality are often copy and pasted between projects, or re-implemented from scratch.
- Over time these features fall out of sync with new features, bug fixes, etc, being addressed in one player but not others.
- Many of these feature have no API docs or guides because they are not part of the players' public APIs

Reasoning

Creates a single place for standards groups to request and oversee feature implementation



Implement Common-Media-Client-Data specification #3176

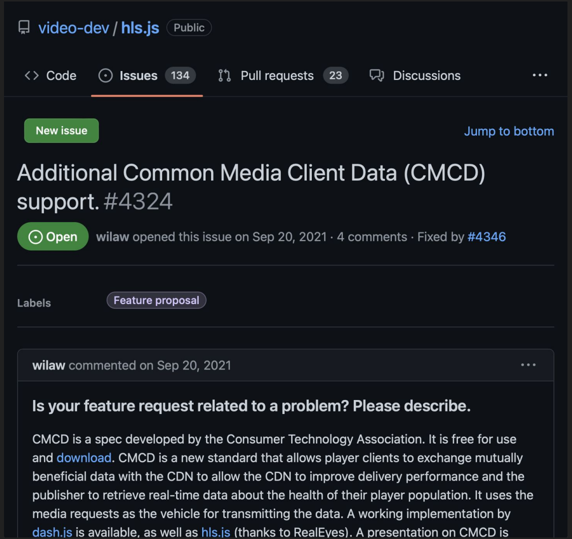
Closed wilaw opened this issue on Feb 4, 2020 · 9 comments

Assignees

Labels: Feature Request

wilaw commented on Feb 4, 2020 Member

The CTA is about to complete the Common-Media-Client-Data specification. This specification details a common query arg (or request header) which is used my media players to convey useful information to CDNs. The draft spec is accessible here: https://docs.google.com/document/d/1S_Dj_aHVnbWnjeJYMfLU1D6tZoqYdgHT1stfznBvxig



Additional Common Media Client Data (CMCD) support. #4324

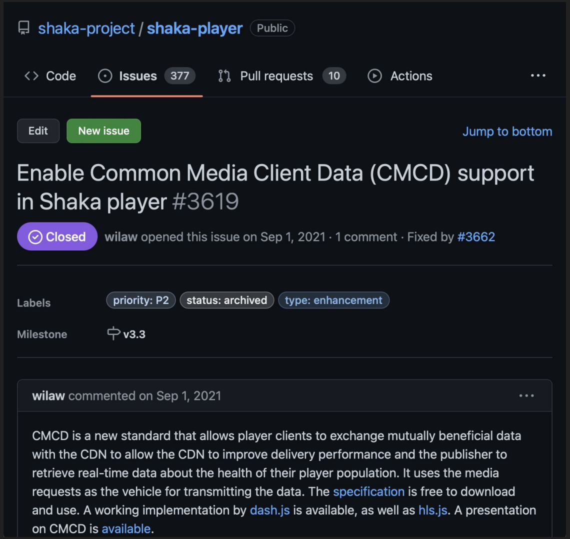
Open wilaw opened this issue on Sep 20, 2021 · 4 comments · Fixed by #4346

Labels: Feature proposal

wilaw commented on Sep 20, 2021

Is your feature request related to a problem? Please describe.

CMCD is a spec developed by the Consumer Technology Association. It is free for use and download. CMCD is a new standard that allows player clients to exchange mutually beneficial data with the CDN to allow the CDN to improve delivery performance and the publisher to retrieve real-time data about the health of their player population. It uses the media requests as the vehicle for transmitting the data. A working implementation by dash.js is available, as well as hls.js (thanks to RealEyes). A presentation on CMCD is



Enable Common Media Client Data (CMCD) support in Shaka player #3619

Closed wilaw opened this issue on Sep 1, 2021 · 1 comment · Fixed by #3662

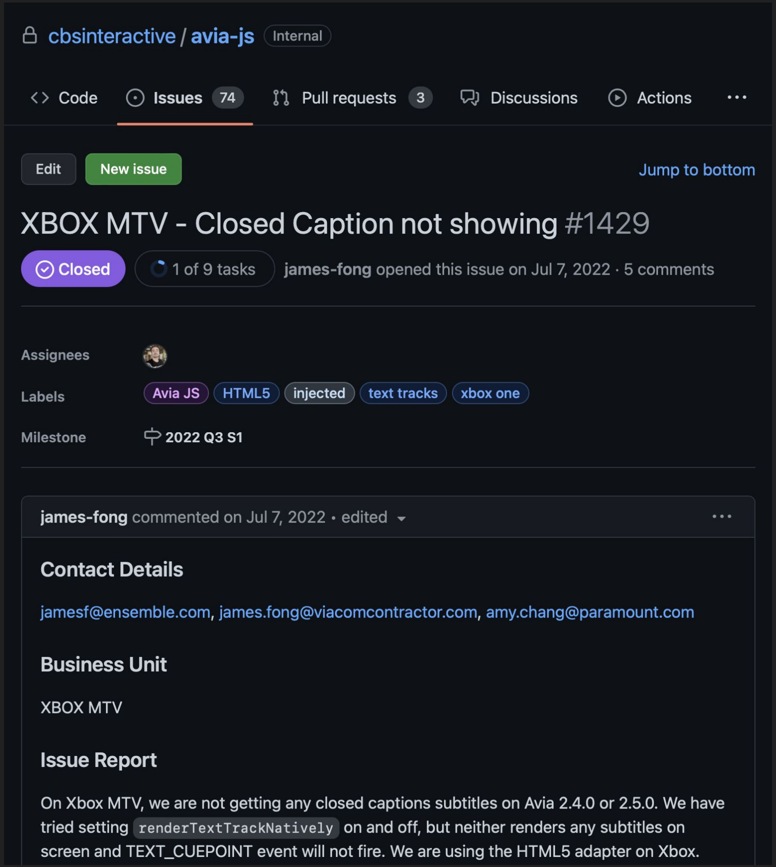
Labels: priority: P2 status: archived type: enhancement

Milestone: v3.3

wilaw commented on Sep 1, 2021

CMCD is a new standard that allows player clients to exchange mutually beneficial data with the CDN to allow the CDN to improve delivery performance and the publisher to retrieve real-time data about the health of their player population. It uses the media requests as the vehicle for transmitting the data. The specification is free to download and use. A working implementation by dash.js is available, as well as hls.js. A presentation on CMCD is available.

Reasoning



XBOX MTV - Closed Caption not showing #1429

Closed · 1 of 9 tasks · **james-fong** opened this issue on Jul 7, 2022 · 5 comments

Assignees

Labels: Avia JS, HTML5, injected, text tracks, xbox one

Milestone: 2022 Q3 S1

james-fong commented on Jul 7, 2022 • edited

Contact Details

jamesf@ensemble.com, james.fong@viacomcontractor.com, amy.chang@paramount.com

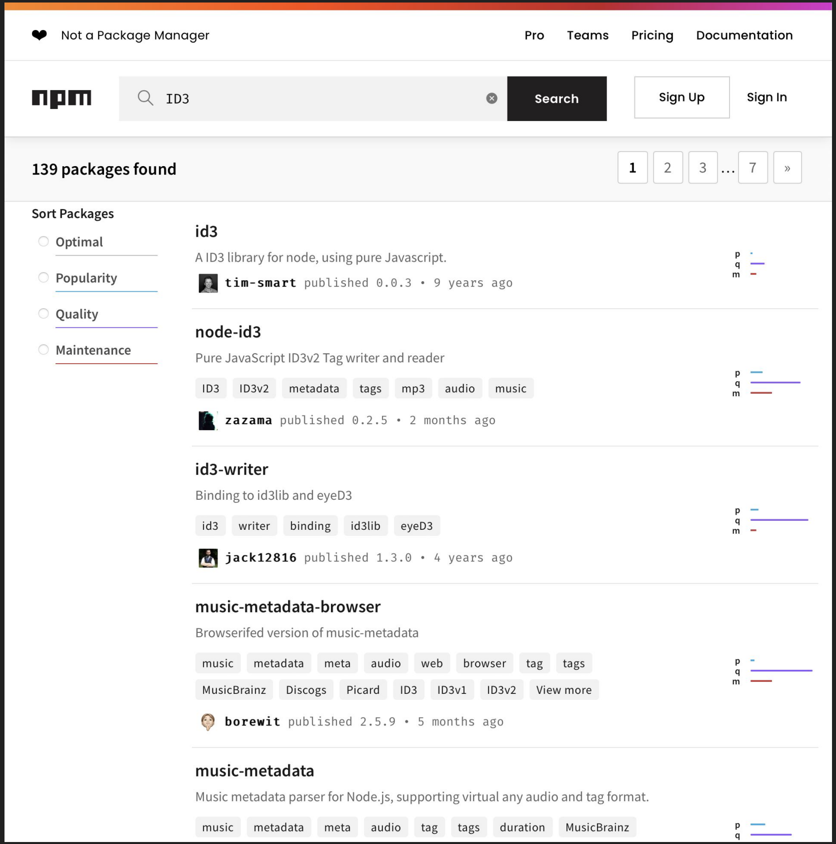
Business Unit

XBOX MTV

Issue Report

On Xbox MTV, we are not getting any closed captions subtitles on Avia 2.4.0 or 2.5.0. We have tried setting `renderTextTrackNatively` on and off, but neither renders any subtitles on screen and TEXT_CUEPOINT event will not fire. We are using the HTML5 adapter on Xbox.

Reasoning



❤ Not a Package Manager

Pro Teams Pricing Documentation

npm

ID3

Search

Sign Up Sign In

139 packages found

1 2 3 … 7 »

Sort Packages

○ Optimal

○ Popularity

○ Quality

○ Maintenance

id3

A ID3 library for node, using pure Javascript.

tim-smart published 0.0.3 • 9 years ago

node-id3

Pure JavaScript ID3v2 Tag writer and reader

ID3 ID3v2 metadata tags mp3 audio music

zazama published 0.2.5 • 2 months ago

id3-writer

Binding to id3lib and eyeD3

id3 writer binding id3lib eyeD3

jack12816 published 1.3.0 • 4 years ago

music-metadata-browser

Browserifed version of music-metadata

music metadata meta audio web browser tag tags MusicBrainz Discogs Picard ID3 ID3v1 ID3v2 View more

borewit published 2.5.9 • 5 months ago

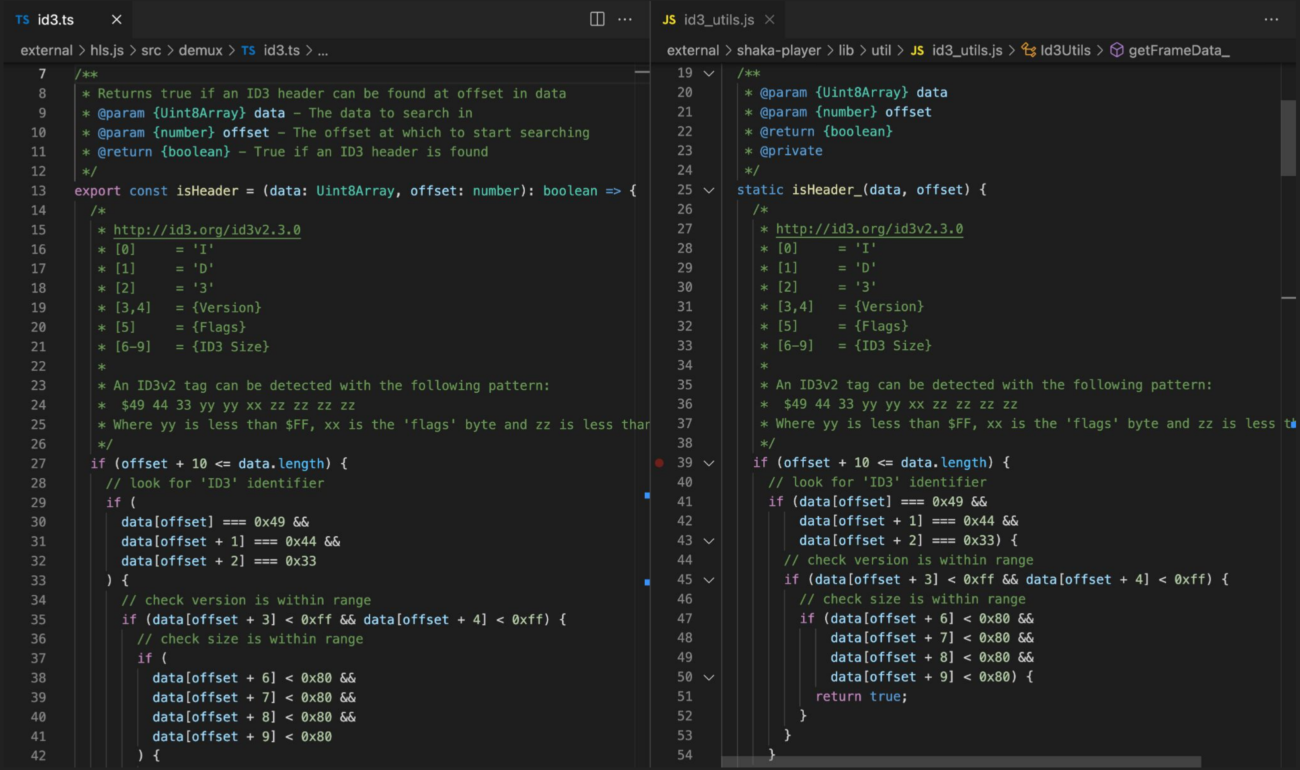
music-metadata

Music metadata parser for Node.js, supporting virtual any audio and tag format.

music metadata meta audio tag tags duration MusicBrainz

SVTA
Streaming Video Technology Alliance

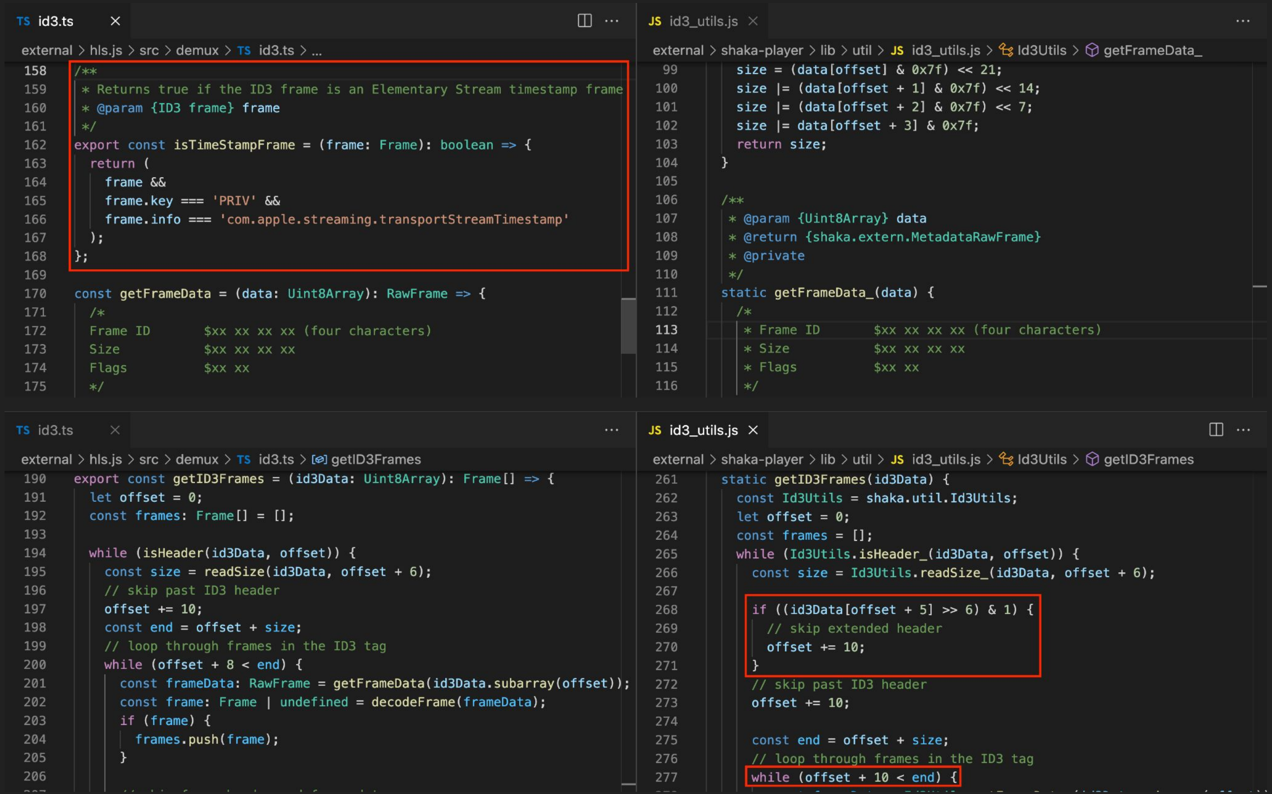
Reasoning


TS id3.ts
external > hls.js > src > demux > TS id3.ts > ...
JS id3_utils.js
external > shaka-player > lib > util > JS id3_utils.js > Id3Utils > getFrameData_

```ts
/**
 * Returns true if an ID3 header can be found at offset in data
 * @param {Uint8Array} data - The data to search in
 * @param {number} offset - The offset at which to start searching
 * @return {boolean} - True if an ID3 header is found
 */
export const isHeader = (data: Uint8Array, offset: number): boolean => {
  /*
   * http://id3.org/id3v2.3.0
   * [0]     = 'I'
   * [1]     = 'D'
   * [2]     = '3'
   * [3,4]   = {Version}
   * [5]     = {Flags}
   * [6-9]   = {ID3 Size}
   *
   * An ID3v2 tag can be detected with the following pattern:
   *  $49 44 33 yy yy xx zz zz zz zz
   * Where yy is less than $FF, xx is the 'flags' byte and zz is less than
   */
  if (offset + 10 <= data.length) {
    // look for 'ID3' identifier
    if (
      data[offset] === 0x49 &&
      data[offset + 1] === 0x44 &&
      data[offset + 2] === 0x33
    ) {
      // check version is within range
      if (data[offset + 3] < 0xff && data[offset + 4] < 0xff) {
        // check size is within range
        if (
          data[offset + 6] < 0x80 &&
          data[offset + 7] < 0x80 &&
          data[offset + 8] < 0x80 &&
          data[offset + 9] < 0x80
        ) {
```

```js
/**
 * @param {Uint8Array} data
 * @param {number} offset
 * @return {boolean}
 * @private
 */
static isHeader_(data, offset) {
  /*
   * http://id3.org/id3v2.3.0
   * [0]     = 'I'
   * [1]     = 'D'
   * [2]     = '3'
   * [3,4]   = {Version}
   * [5]     = {Flags}
   * [6-9]   = {ID3 Size}
   *
   * An ID3v2 tag can be detected with the following pattern:
   *  $49 44 33 yy yy xx zz zz zz zz
   * Where yy is less than $FF, xx is the 'flags' byte and zz is less t
   */
  if (offset + 10 <= data.length) {
    // look for 'ID3' identifier
    if (data[offset] === 0x49 &&
        data[offset + 1] === 0x44 &&
        data[offset + 2] === 0x33) {
      // check version is within range
      if (data[offset + 3] < 0xff && data[offset + 4] < 0xff) {
        // check size is within range
        if (data[offset + 6] < 0x80 &&
            data[offset + 7] < 0x80 &&
            data[offset + 8] < 0x80 &&
            data[offset + 9] < 0x80) {
          return true;
        }
      }
    }
```

Reasoning



TS id3.ts
external > hls.js > src > demux > TS id3.ts > ...

```
/**
 * Returns true if the ID3 frame is an Elementary Stream timestamp frame
 * @param {ID3 frame} frame
 */
export const isTimeStampFrame = (frame: Frame): boolean => {
  return (
    frame &&
    frame.key === 'PRIV' &&
    frame.info === 'com.apple.streaming.transportStreamTimestamp'
  );
};

const getFrameData = (data: Uint8Array): RawFrame => {
  /*
  Frame ID       $xx xx xx xx (four characters)
  Size           $xx xx xx xx
  Flags          $xx xx
  */
```

JS id3_utils.js
external > shaka-player > lib > util > JS id3_utils.js > Id3Utils > getFrameData_

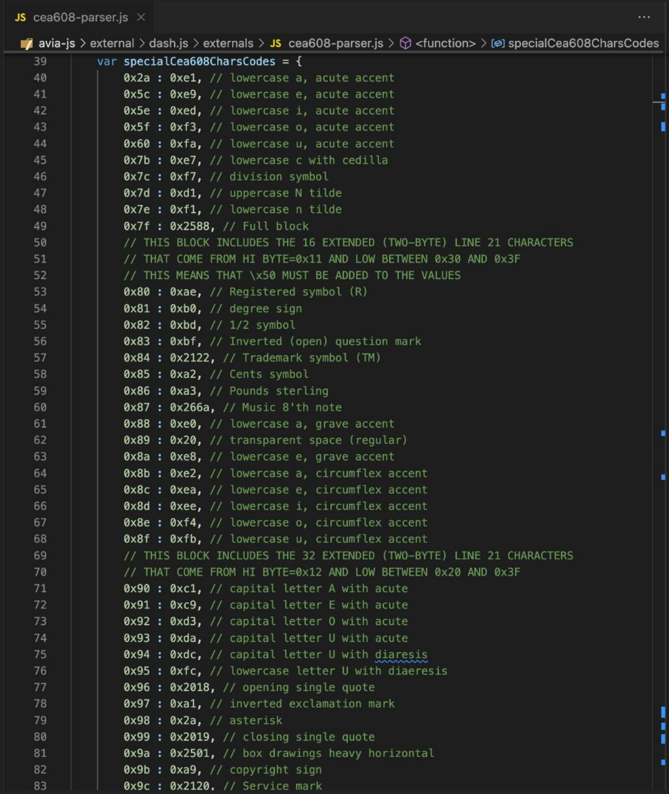
```
    size = (data[offset] & 0x7f) << 21;
    size |= (data[offset + 1] & 0x7f) << 14;
    size |= (data[offset + 2] & 0x7f) << 7;
    size |= data[offset + 3] & 0x7f;
    return size;
  }

  /**
   * @param {Uint8Array} data
   * @return {shaka.extern.MetadataRawFrame}
   * @private
   */
  static getFrameData_(data) {
    /*
     * Frame ID       $xx xx xx xx (four characters)
     * Size           $xx xx xx xx
     * Flags          $xx xx
     */
```

```
export const getID3Frames = (id3Data: Uint8Array): Frame[] => {
  let offset = 0;
  const frames: Frame[] = [];

  while (isHeader(id3Data, offset)) {
    const size = readSize(id3Data, offset + 6);
    // skip past ID3 header
    offset += 10;
    const end = offset + size;
    // loop through frames in the ID3 tag
    while (offset + 8 < end) {
      const frameData: RawFrame = getFrameData(id3Data.subarray(offset));
      const frame: Frame | undefined = decodeFrame(frameData);
      if (frame) {
        frames.push(frame);
      }

```

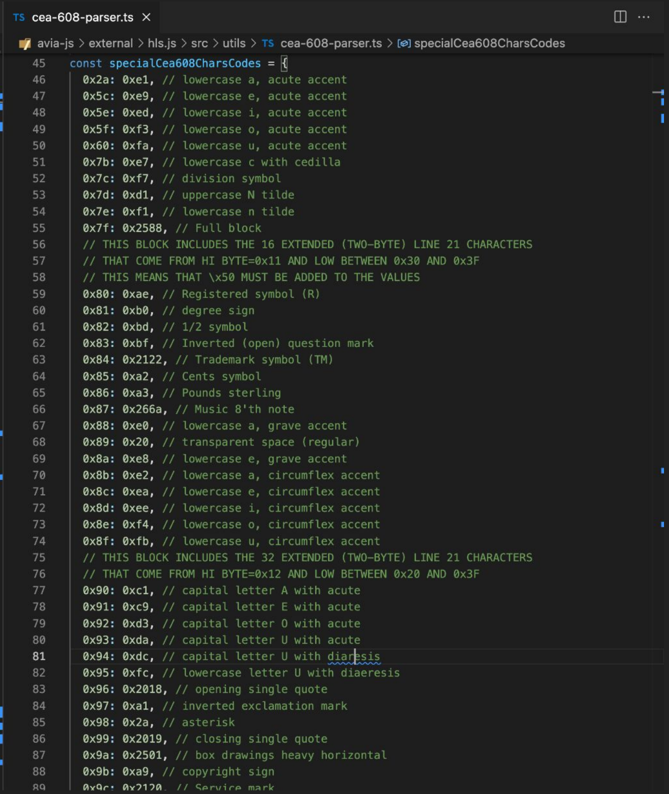
JS id3_utils.js
external > shaka-player > lib > util > JS id3_utils.js > Id3Utils > getID3Frames

```
  static getID3Frames(id3Data) {
    const Id3Utils = shaka.util.Id3Utils;
    let offset = 0;
    const frames = [];
    while (Id3Utils.isHeader_(id3Data, offset)) {
      const size = Id3Utils.readSize_(id3Data, offset + 6);

      if ((id3Data[offset + 5] >> 6) & 1) {
        // skip extended header
        offset += 10;
      }
      // skip past ID3 header
      offset += 10;

      const end = offset + size;
      // loop through frames in the ID3 tag
      while (offset + 10 < end) {
```

Reasoning

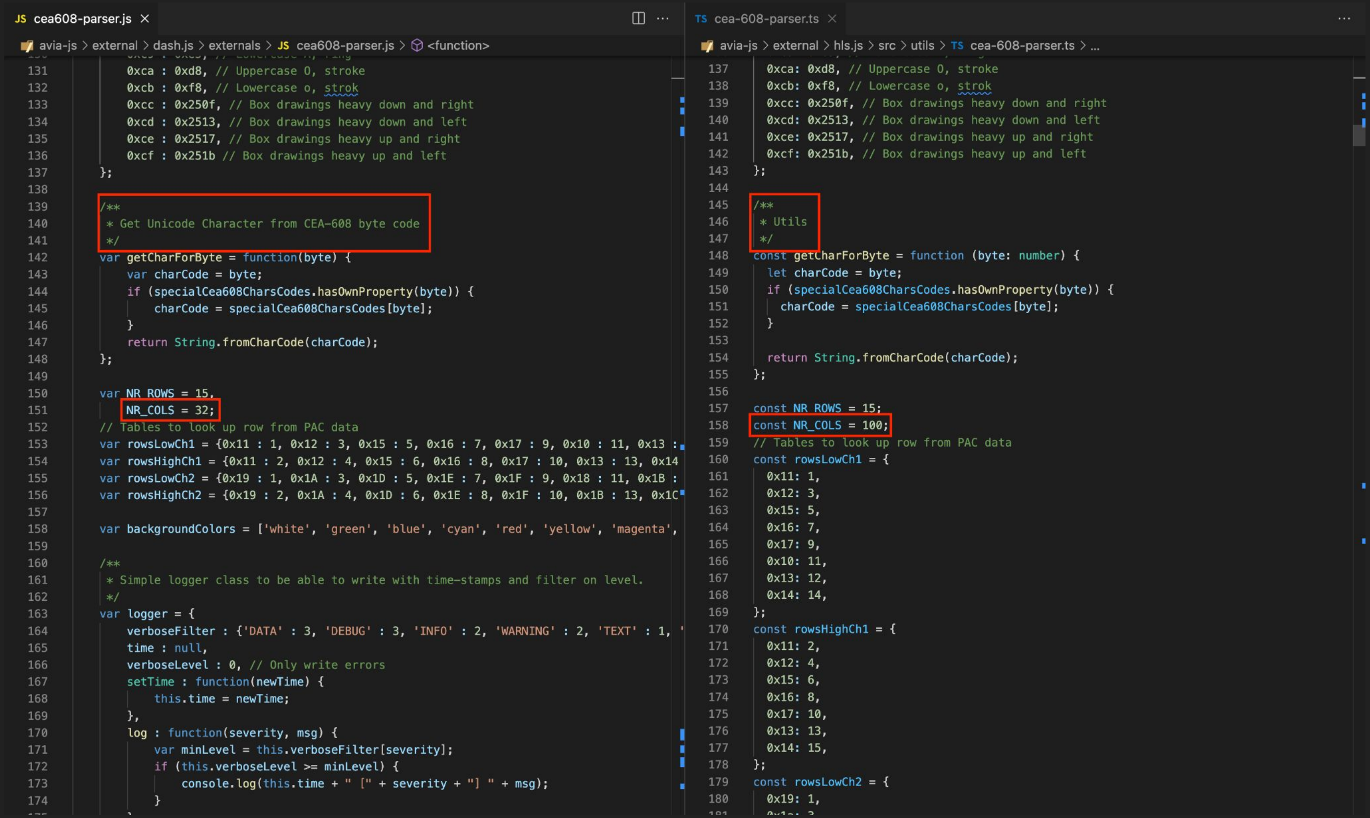

cea608-parser.js
avia-js > external > dash.js > externals > cea608-parser.js > <function> > specialCea608CharsCodes

```
    var specialCea608CharsCodes = {
        0x2a : 0xe1, // lowercase a, acute accent
        0x5c : 0xe9, // lowercase e, acute accent
        0x5e : 0xed, // lowercase i, acute accent
        0x5f : 0xf3, // lowercase o, acute accent
        0x60 : 0xfa, // lowercase u, acute accent
        0x7b : 0xe7, // lowercase c with cedilla
        0x7c : 0xf7, // division symbol
        0x7d : 0xd1, // uppercase N tilde
        0x7e : 0xf1, // lowercase n tilde
        0x7f : 0x2588, // Full block
        // THIS BLOCK INCLUDES THE 16 EXTENDED (TWO-BYTE) LINE 21 CHARACTERS
        // THAT COME FROM HI BYTE=0x11 AND LOW BETWEEN 0x30 AND 0x3F
        // THIS MEANS THAT \x50 MUST BE ADDED TO THE VALUES
        0x80 : 0xae, // Registered symbol (R)
        0x81 : 0xb0, // degree sign
        0x82 : 0xbd, // 1/2 symbol
        0x83 : 0xbf, // Inverted (open) question mark
        0x84 : 0x2122, // Trademark symbol (TM)
        0x85 : 0xa2, // Cents symbol
        0x86 : 0xa3, // Pounds sterling
        0x87 : 0x266a, // Music 8'th note
        0x88 : 0xe0, // lowercase a, grave accent
        0x89 : 0x20, // transparent space (regular)
        0x8a : 0xe8, // lowercase e, grave accent
        0x8b : 0xe2, // lowercase a, circumflex accent
        0x8c : 0xea, // lowercase e, circumflex accent
        0x8d : 0xee, // lowercase i, circumflex accent
        0x8e : 0xf4, // lowercase o, circumflex accent
        0x8f : 0xfb, // lowercase u, circumflex accent
        // THIS BLOCK INCLUDES THE 32 EXTENDED (TWO-BYTE) LINE 21 CHARACTERS
        // THAT COME FROM HI BYTE=0x12 AND LOW BETWEEN 0x20 AND 0x3F
        0x90 : 0xc1, // capital letter A with acute
        0x91 : 0xc9, // capital letter E with acute
        0x92 : 0xd3, // capital letter O with acute
        0x93 : 0xda, // capital letter U with acute
        0x94 : 0xdc, // capital letter U with diaresis
        0x95 : 0xfc, // lowercase letter U with diaeresis
        0x96 : 0x2018, // opening single quote
        0x97 : 0xa1, // inverted exclamation mark
        0x98 : 0x2a, // asterisk
        0x99 : 0x2019, // closing single quote
        0x9a : 0x2501, // box drawings heavy horizontal
        0x9b : 0xa9, // copyright sign
        0x9c : 0x2120, // Service mark
```


cea-608-parser.ts
avia-js > external > hls.js > src > utils > cea-608-parser.ts > specialCea608CharsCodes

```
const specialCea608CharsCodes = {
  0x2a: 0xe1, // lowercase a, acute accent
  0x5c: 0xe9, // lowercase e, acute accent
  0x5e: 0xed, // lowercase i, acute accent
  0x5f: 0xf3, // lowercase o, acute accent
  0x60: 0xfa, // lowercase u, acute accent
  0x7b: 0xe7, // lowercase c with cedilla
  0x7c: 0xf7, // division symbol
  0x7d: 0xd1, // uppercase N tilde
  0x7e: 0xf1, // lowercase n tilde
  0x7f: 0x2588, // Full block
  // THIS BLOCK INCLUDES THE 16 EXTENDED (TWO-BYTE) LINE 21 CHARACTERS
  // THAT COME FROM HI BYTE=0x11 AND LOW BETWEEN 0x30 AND 0x3F
  // THIS MEANS THAT \x50 MUST BE ADDED TO THE VALUES
  0x80: 0xae, // Registered symbol (R)
  0x81: 0xb0, // degree sign
  0x82: 0xbd, // 1/2 symbol
  0x83: 0xbf, // Inverted (open) question mark
  0x84: 0x2122, // Trademark symbol (TM)
  0x85: 0xa2, // Cents symbol
  0x86: 0xa3, // Pounds sterling
  0x87: 0x266a, // Music 8'th note
  0x88: 0xe0, // lowercase a, grave accent
  0x89: 0x20, // transparent space (regular)
  0x8a: 0xe8, // lowercase e, grave accent
  0x8b: 0xe2, // lowercase a, circumflex accent
  0x8c: 0xea, // lowercase e, circumflex accent
  0x8d: 0xee, // lowercase i, circumflex accent
  0x8e: 0xf4, // lowercase o, circumflex accent
  0x8f: 0xfb, // lowercase u, circumflex accent
  // THIS BLOCK INCLUDES THE 32 EXTENDED (TWO-BYTE) LINE 21 CHARACTERS
  // THAT COME FROM HI BYTE=0x12 AND LOW BETWEEN 0x20 AND 0x3F
  0x90: 0xc1, // capital letter A with acute
  0x91: 0xc9, // capital letter E with acute
  0x92: 0xd3, // capital letter O with acute
  0x93: 0xda, // capital letter U with acute
  0x94: 0xdc, // capital letter U with diaresis
  0x95: 0xfc, // lowercase letter U with diaeresis
  0x96: 0x2018, // opening single quote
  0x97: 0xa1, // inverted exclamation mark
  0x98: 0x2a, // asterisk
  0x99: 0x2019, // closing single quote
  0x9a: 0x2501, // box drawings heavy horizontal
  0x9b: 0xa9, // copyright sign
  0x9c: 0x2120, // Service mark
```

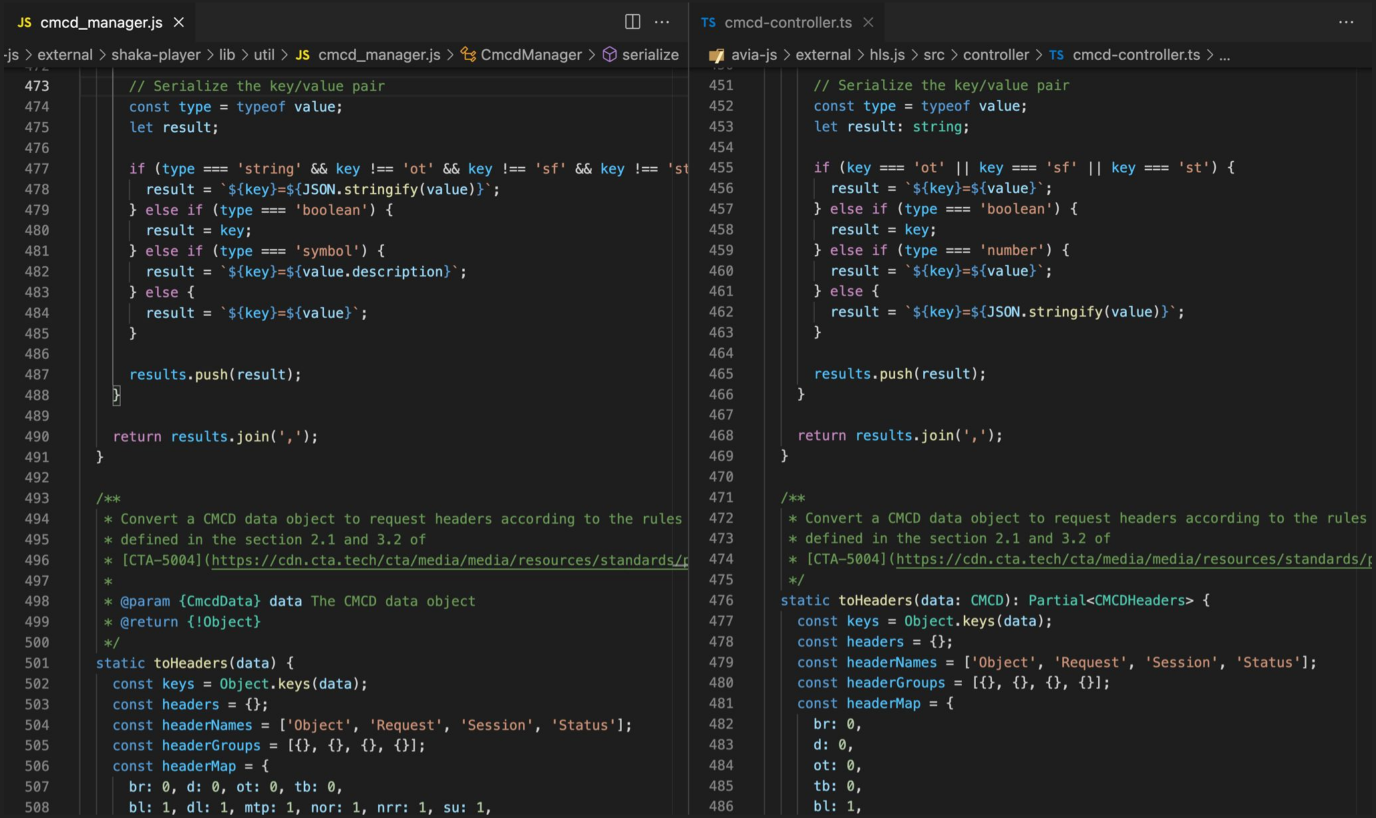
Reasoning


JS cea608-parser.js
avia-js > external > dash.js > externals > JS cea608-parser.js > <function>
TS cea-608-parser.ts
avia-js > external > hls.js > src > utils > TS cea-608-parser.ts > ...

```javascript
    0xca : 0xd8, // Uppercase O, stroke
    0xcb : 0xf8, // Lowercase o, strok
    0xcc : 0x250f, // Box drawings heavy down and right
    0xcd : 0x2513, // Box drawings heavy down and left
    0xce : 0x2517, // Box drawings heavy up and right
    0xcf : 0x251b // Box drawings heavy up and left
};

/**
 * Get Unicode Character from CEA-608 byte code
 */
var getCharForByte = function(byte) {
    var charCode = byte;
    if (specialCea608CharsCodes.hasOwnProperty(byte)) {
        charCode = specialCea608CharsCodes[byte];
    }
    return String.fromCharCode(charCode);
};

var NR_ROWS = 15,
    NR_COLS = 32;
// Tables to look up row from PAC data
var rowsLowCh1 = {0x11 : 1, 0x12 : 3, 0x15 : 5, 0x16 : 7, 0x17 : 9, 0x10 : 11, 0x13 :
var rowsHighCh1 = {0x11 : 2, 0x12 : 4, 0x15 : 6, 0x16 : 8, 0x17 : 10, 0x13 : 13, 0x14
var rowsLowCh2 = {0x19 : 1, 0x1A : 3, 0x1D : 5, 0x1E : 7, 0x1F : 9, 0x18 : 11, 0x1B :
var rowsHighCh2 = {0x19 : 2, 0x1A : 4, 0x1D : 6, 0x1E : 8, 0x1F : 10, 0x1B : 13, 0x1C

var backgroundColors = ['white', 'green', 'blue', 'cyan', 'red', 'yellow', 'magenta',

/**
 * Simple logger class to be able to write with time-stamps and filter on level.
 */
var logger = {
    verboseFilter : {'DATA' : 3, 'DEBUG' : 3, 'INFO' : 2, 'WARNING' : 2, 'TEXT' : 1, '
    time : null,
    verboseLevel : 0, // Only write errors
    setTime : function(newTime) {
        this.time = newTime;
    },
    log : function(severity, msg) {
        var minLevel = this.verboseFilter[severity];
        if (this.verboseLevel >= minLevel) {
            console.log(this.time + " [" + severity + "] " + msg);
        }
```

```typescript
  0xca: 0xd8, // Uppercase O, stroke
  0xcb: 0xf8, // Lowercase o, strok
  0xcc: 0x250f, // Box drawings heavy down and right
  0xcd: 0x2513, // Box drawings heavy down and left
  0xce: 0x2517, // Box drawings heavy up and right
  0xcf: 0x251b, // Box drawings heavy up and left
};

/**
 * Utils
 */
const getCharForByte = function (byte: number) {
  let charCode = byte;
  if (specialCea608CharsCodes.hasOwnProperty(byte)) {
    charCode = specialCea608CharsCodes[byte];
  }

  return String.fromCharCode(charCode);
};

const NR_ROWS = 15;
const NR_COLS = 100;
// Tables to look up row from PAC data
const rowsLowCh1 = {
  0x11: 1,
  0x12: 3,
  0x15: 5,
  0x16: 7,
  0x17: 9,
  0x10: 11,
  0x13: 12,
  0x14: 14,
};
const rowsHighCh1 = {
  0x11: 2,
  0x12: 4,
  0x15: 6,
  0x16: 8,
  0x17: 10,
  0x13: 13,
  0x14: 15,
};
const rowsLowCh2 = {
  0x19: 1,
```

Reasoning


JS cmcd_manager.js ×
-js > external > shaka-player > lib > util > JS cmcd_manager.js > CmcdManager > serialize
TS cmcd-controller.ts ×
avia-js > external > hls.js > src > controller > TS cmcd-controller.ts > ...

```js
      // Serialize the key/value pair
      const type = typeof value;
      let result;

      if (type === 'string' && key !== 'ot' && key !== 'sf' && key !== 'st
        result = `${key}=${JSON.stringify(value)}`;
      } else if (type === 'boolean') {
        result = key;
      } else if (type === 'symbol') {
        result = `${key}=${value.description}`;
      } else {
        result = `${key}=${value}`;
      }

      results.push(result);
    }

    return results.join(',');
  }

  /**
   * Convert a CMCD data object to request headers according to the rules
   * defined in the section 2.1 and 3.2 of
   * [CTA-5004](https://cdn.cta.tech/cta/media/media/resources/standards/p
   *
   * @param {CmcdData} data The CMCD data object
   * @return {!Object}
   */
  static toHeaders(data) {
    const keys = Object.keys(data);
    const headers = {};
    const headerNames = ['Object', 'Request', 'Session', 'Status'];
    const headerGroups = [{}, {}, {}, {}];
    const headerMap = {
      br: 0, d: 0, ot: 0, tb: 0,
      bl: 1, dl: 1, mtp: 1, nor: 1, nrr: 1, su: 1,
```

```ts
      // Serialize the key/value pair
      const type = typeof value;
      let result: string;

      if (key === 'ot' || key === 'sf' || key === 'st') {
        result = `${key}=${value}`;
      } else if (type === 'boolean') {
        result = key;
      } else if (type === 'number') {
        result = `${key}=${value}`;
      } else {
        result = `${key}=${JSON.stringify(value)}`;
      }

      results.push(result);
    }

    return results.join(',');
  }

  /**
   * Convert a CMCD data object to request headers according to the rules
   * defined in the section 2.1 and 3.2 of
   * [CTA-5004](https://cdn.cta.tech/cta/media/media/resources/standards/p
   */
  static toHeaders(data: CMCD): Partial<CMCDHeaders> {
    const keys = Object.keys(data);
    const headers = {};
    const headerNames = ['Object', 'Request', 'Session', 'Status'];
    const headerGroups = [{}, {}, {}, {}];
    const headerMap = {
      br: 0,
      d: 0,
      ot: 0,
      tb: 0,
      bl: 1,
```

Current Status

- The repository is now public
  - Github: https://github.com/streaming-video-technology-alliance/common-media-library
  - NPM: @svta/common-media-library
  - Docs: https://streaming-video-technology-alliance.github.io/common-media-library/
- Current libraries include:
  - CMCD - addition of Structured fields values sets us up for upcoming v2
  - CMSD
  - ID3 parsing
  - Common Media Request/Response interfaces - helps standardize CDN load balancer implementations across players
- The library is now integrated in the following projects:
  - hls.js 1.5.0 - CMCD
  - dash.js 5.0.0 - CMCD
  - @montevideo-tech/videojs-cmcd 1.0.6 - video.js CMCD plugin

Next Steps

- Integrate the library into new players
  - shaka-player: CMCD, CMSD, ID3
- Integrate more CML utilities in current players
  - ID3 - hls.js, dash.js
  - CMSD - dash.js
- Implement new features / utilities
  - ISO Box Parsing
  - Caption rendering
- Open call for new features
  - https://github.com/streaming-video-technology-alliance/common-media-library/discussions/54

Thank You

Contact

Email: casey.occhialini@paramount.com

Linkedin: https://www.linkedin.com/in/caseyocchialini/

Github: @littlespex