**3GPP TSG-S4 Meeting #127-bis-e*****S4-240545***

**Electronic, , 8th–12th April 2024**

|  |
| --- |
| *CR-Form-v12.0* |
| **CHANGE REQUEST** |
|  |
|  | **26.512** | **CR** | **0066** | **rev** |  | **Current version:** | **18.1.0** |  |
|  |
| *For* [***HE******LP***](http://www.3gpp.org/3G_Specs/CRs.htm#_blank)*on using this form: comprehensive instructions can be found at* [*http://www.3gpp.org/Change-Requests*](http://www.3gpp.org/Change-Requests)*.* |
|  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Proposed change affects:*** | UICC apps |  | ME | **X** | Radio Access Network |  | Core Network |  |

|  |
| --- |
|  |
| ***Title:***  |  |
|  |  |
| ***Source to WG:*** | BBC |
| ***Source to TSG:*** | S4 |
|  |  |
| ***Work item code:*** | 5GMS\_Pro\_Ph2 |  | ***Date:*** | 2024-03-28 |
|  |  |  |  |  |
| ***Category:*** | **B** |  | ***Release:*** | Rel-18 |
|  | *Use one of the following categories:****F*** *(correction)****A*** *(mirror corresponding to a change in an earlier release)****B*** *(addition of feature),* ***C*** *(functional modification of feature)****D*** *(editorial modification)*Detailed explanations of the above categories canbe found in 3GPP [TR 21.900](http://www.3gpp.org/ftp/Specs/html-info/21900.htm). | *Use one of the following releases:Rel-8 (Release 8)Rel-9 (Release 9)Rel-10 (Release 10)Rel-11 (Release 11)…Rel-15 (Release 15)Rel-16 (Release 16)Rel-17 (Release 17)Rel-18 (Release 18)* |
|  |  |
| ***Reason for change:*** | TS 26.510 adds the concept of a media delivery session identifier to interactions at reference points M6/M11, but TS 26.512 does not yet provide anything equivalent at reference points M7/M11. |
|  |  |
| ***Summary of change:*** | * Add *media delivery session identifier* as a parameter to all methods in clause 13.2.3.
* Add *media delivery session identifier* as a parameter to all notifications in clause 13.2.5.
 |
|  |  |
| ***Consequences if not approved:*** | Entities interacting with the Media Session Handler are not able to unambigously refer to a particular media delivery session. |
| ***Q*** |  |
| ***Clauses affected:*** | 2, 6.2.3.6 (new), 10, 10.1, 10.1A (new), 13.1, 13.2.1, 13.2.3, 13.2.3.1, 13.2.3.2, 13.2.3.3, 13.2.3.4, 13.2.3.5, 13.2.3.6, 13.2.3.7, 13.2.3.8, 13.2.3.9, 13.2.4, 13.2.5, 13.2.6, 13.3.7 |
|  |  |
|  | **Y** | **N** |  |  |
| ***Other specs*** | **X** |  |  Other core specifications | TS 26.510 V2.0.0 |
| ***affected:*** |  | **X** |  Test specifications |  |
| ***(show related CRs)*** |  | **X** |  O&M Specifications |  |
|  |  |
| ***Other comments:*** |  |
|  |  |
| ***This CR's revision history:*** | CR0066 [S4-240545]: Submitted for WG endorsement. |

FIRST CHANGE

# 2 References

[1] 3GPP TR 21.905: “Vocabulary for 3GPP Specifications”.

…

[56] 3GPP TS 26.510: "Media delivery; interactions and APIs for provisioning and media session handling".

Next change

#### 6.2.3.6 Media delivery session identifier

When required, the media delivery session identifier specified in clause 7.3.2 of TS 26.510 [56] shall be conveyed in the HTTP header 3gpp-Media-Delivery-Session-Id.

EXAMPLE: 3gpp-Media-Delivery-Session-Id: 8bf9f090-82fd-4686-aa4a-39e6a9381b76

Next change

# 10 Media Streaming (M4) interface

## 10.1 General

This clause specifies the interface for downlink and uplink media streaming at reference point M4 using different distribution formats and protocols. TS 26.511 [35] defines the integration of several media codecs into 5G Media Streaming, and provides requirements and recommendations for the support of these media profiles in specific 5G Media Streaming profiles. However, 5GMS is not restricted to the media profiles defined in TS 26.511 [35]: any CMAF media profile that integrates with the APIs specified in the present document may be used for media streaming at this reference point.

## 10.1A Media delivery session identification

All requests addressed by the Media Stream Handler (Media Player or Media Streamer) to the 5GMS AS at reference point M4 should cite a media delivery session identifier using the HTTP header specified in clause 6.2.3.6. The value of this identifier shall be nominated by the Media Session Handler and shall be conveyed to the Media Stream Handler as specified in clause 13.2.3.2 when a new media streaming session is initiated.

NEXT CHANGE

# 13 UE Media Stream Handler (M7/M11) APIs

## 13.1 General

This clause defines a set of APIs and methods that permit a 5GMS-Aware Application at reference point M7 or a Media Session Handler at reference point M11 to communicate with a Media Stream Handler (Media Player or Media Streamer). The main focus of this clause is to formalize and harmonize commonly available proprietary APIs in order to support the usage of a Media Player or a Media Streamer in a 5G Media Streaming context.

The APIs specified in this clause are language- and runtime-independent. Implementations are expected to provide language bindings appropriate to the UE runtime environment.

## 13.2 DASH Media Player APIs and functions

### 13.2.1 Overview

In the following, it is assumed that the Media Player (in this case a DASH client) adheres to a basic set of functionalities as shown in figure 13.2-1. The DASH client downloads, processes and presents a DASH Media Presentation under the control of a 5GMSd-Aware Application via reference point M7d or of the Media Session Handler via reference point M11d.

The 5GMSd-Aware Application may, in addition, configure the presentation of the media, receive notifications on events, or query the internal status of the DASH Player, also supported through reference point M7d. Different functions of the DASH Access Client that are typically necessary to process a DASH Media Presentation, are shown in Figure 13.2-1. Additional functions may be available as well.



Figure 13.2.1-1: Architecture of DASH-based 5GMSd Client

The key functionalities of each of the functions as shown in figure 13.2-1 are summarized in the following:

- *5GMSd-Aware Application:* Application that makes use of the DASH-based Media Player to play back a DASH Media Presentation using the APIs defined in this clause.

- *Media Player:* A complete player for the playback of a Media Presentation, including the Media Playback and Content Decryption Platform as defined in TS 26.511 [35].

- *Access Client:* A part of the DASH Player that accesses and downloads of the resources and provides the downloaded resources to the Media Playback Platform and Content Decryption for the playback of DASH content.

- *Management:* Controls all internal processes and the communication with the 5GMSd-aware application. In particular this includes the handling of service descriptions and operation points.

- *MPD Processing:* parses and processes the MPD and extracts the relevant information.

- *Adaptation Set Selection:* selects the Adaptation Set based on user, application and/or device capability information. Information provided through M7d may be used.

- *ABR Controller and Dynamic Switching:* runs adaptive bit rate logic and triggers adaptive switching of Representations. Information provided to the DASH client through M7d may be used.

- *Throughput Estimation:* estimates the throughput from the 5GMSd Application Server.

- *Metrics Logging:* logs relevant low-level metrics and provides those to the metrics aggregation and reporting functions in the Media Session Handler.

- *Media Playback Management and Protection Controller:* manages the media playback by moving downloaded information into media playback platform and also addresses handling of protection and DRM related information.

- *Media Playback and Content Decryption Platform:* plays back CMAF-based media content according to the playback requirements in TS 26.511 [35]. It also provides status information as well as events that maybe be provided through M7d.

- *Event Processing:* Processes DASH events and provides information to the 5GMSd-Aware Application as defined in TS 26.247 [4].

This clause focuses on interactions with the Media Player through reference point M7d. In particular, the following aspects of the API are defined:

1) Methods to interact with the Media Player at this reference point are defined in clause 13.2.3.

2) Notification and Error Events raised by the Media Player at this reference point are defined in clause 13.2.4.

3) Configuration and Settings of the Media Player at this reference point are defined in clause 13.2.5.

4) Status Information exposed by the Media Player at this reference point is defined in clause 13.2.6.

Communication between the Access Client and the media playback platform of the Media Player is defined in TS 26.511 [35].

A 5GMSd Client for DASH distribution shall support the APIs defined in this clause 13.

NOTE: The initial APIs have largely been designed based on the dash.js APIs documented here: <http://cdn.dashjs.org/latest/jsdoc>.

### 13.2.2 Media Player model

Figure 13.2.2-1 provides an informative client state model in order to appropriately describe the messages on the Media streaming service API. Six different states are defined.

State changes may happen based on:

- Calls from application.

- Information provided in the Media Presentation Description (MPD).



Figure 13.2.2-1: State Diagram for Media Player

Table 13.2.2-1 defines states for the Media Player. Detailed descriptions are provided in the following clauses.

Table 13.2.2-1: States of Media Player

| States | Definition |
| --- | --- |
| IDLE | The Media Player is not associated with any application. |
| INITIALIZED | The Media Player is associated with an application and the M7d API communication is established. |
| READY | The Media Player has loaded an MPD and is able to playback the media in this Media Presentation. It also updates the MPD according to the MPD update mechanism. |
| PRELOADED | The Media Player has pre-loaded all media information in order to start playback instantaneously. It also updates the MPD according to the MPD update mechanism. |
| PLAYING | The Media Player is playing the Media Presentation. It also updates the MPD according to the MPD update mechanism. |
| PAUSED | The playback of the Media Presentation is paused. It also updates the MPD according to the MPD update mechanism. |

It is assumed that the DASH Access Client manages the playback of at most one CMAF track for each media type, namely one for video, one for audio and one for subtitles as defined in TS 26.511 [35]. Playback of multiple CMAF tracks of the same media type is not excluded by the present document, but details are for further study.

### 13.2.3 Media Player methods

#### 13.2.3.1 General

Based on the state model in clause 13.2.2, this clause introduces relevant procedures and API calls exposed by the Media Player to the 5GMSd-Aware Application at reference point M7d and to the Media Session Handler at reference point M11d.

Table 13.2.3.1-1 provides an overview over the methods defined for the DASH-based streaming API. Note that in implementations, additional methods may be supported.

Table 13.2.3.1-1: Methods defined for DASH Streaming API

| Method | State after success | Brief description | Clause |
| --- | --- | --- | --- |
| initialize() | INITIALIZED | The Media Player is created. | 13.2.3.2 |
| attach() | READY | Sets a source URL to an MPD file or a previously downloaded and parsed MPD. | 13.2.3.3 |
| preload() | PRELOADED | Streaming the media is initiated. | 13.2.3.4 |
| play() | PLAYING | Playback of the media is initiated. | 13.2.3.5 |
| pause() | PAUSED | Playback of the media is paused. | 13.2.3.6 |
| seek() | PLAYING | The playback time of the media is altered. | 13.2.3.7 |
| reset() | INITIALIZED | All media related information is reset. | 13.2.3.8 |
| destroy() | IDLE | All media player related information is reset and API communication is stopped. | 13.2.3.9 |

#### 13.2.3.2 Initialize

This clause defines the initialize() method.

A new Media Player instance is created using the initialize() method. The following functions are initialized:

- Media Playback Management in order to enable API-based communication through M7d and/or M11d. In particular, the *Notifications and Errors API* (see clause 13.2.5) and the *Status Information* (see clause 13.2.6) are established.

The parameters of the method are defined in table 13.2.3.2-1.

Table 13.2.3.2-1: Parameters of initialize() method

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| sessionId | string | A media delivery session identifier as specified in clause 7.3.2 of TS 26.510 [56]. |

#### 13.2.3.3 Attach

This clause defines the attach() method.

The following pre-conditions apply:

- The Media Player is in INITIALIZED state.

An 5GMSd-Aware Application calls attachMPD() to set a source URL to an MPD file or a previously downloaded and parsed MPD.

The parameters of the method are defined in table 13.2.3.3-1.

Table 13.2.3.3-1: Parameters of attachMPD() method

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| sessionId | string | The media delivery session identifier (as specified in clause 7.3.2 of TS 26.510 [56]) of an initialised downlink media streaming session. |
| urlOrMPD | string | Object | A URL to a valid MPD or a valid MPD as defined in ISO/IEC 23009-1 [32] or TS 26.247 [4].The URL may be augmented by MPD Anchors as defined in clause C.4 of ISO/IEC 23009‑1 [32]. |

The following Media Player Actions are expected:

- The *Request Scheduling* and *Download* functions are established.

- If the input is a URL, the Media Player requests the MPD at the corresponding URL through reference point M4d.

- If the MPD is not found after multiple retries, an error ERROR\_MPD\_NOT\_FOUND is returned and the process is terminated.

- The *MPD Processing* function is established and the MPD parsed.

- If the MPD is not valid, an error ERROR\_MPD\_NOT\_VALID is returned and the process is terminated.

- If the DASH Player does not support the profiles as indicated in the MPD, an error ERROR\_PROFILE\_NOT\_SUPPORTED is returned and the process is terminated.

- Depending on the type of the MPD, possibly present anchors as well as the wall-clock time, the Media Player selects the Period in the content that is expected to be played next.

- The *Media Playback Management and Protection Controller* is established.

- The MPD is parsed for available Service Descriptions (including Media Subsets and Adaptation Sets). By using capability mechanisms defined in TS 26.511 [35] as well as using other information (language settings, output capabilities, accessibility settings), the Media Player identifies a set of permissible Service Descriptions including Media Subsets and Adaptation Sets. If no Adaptation Sets are capable to be played, an error ERROR\_MEDIA\_NOT\_SUPPORTED is returned and the process is terminated.

- The available Service Descriptions including included Adaptation Sets are provided to the 5GMS-Aware Application through reference point M7d.

- The application may select a Service Description instance as well as Adaptation Sets. Additional Service Descriptions parameters may be configured by the 5GMSd-Aware Application through reference point M7d.

- Based on the service description parameters and selected Adaptation Sets:

- the Operation Point parameters are set.

- the *Media Playback Platform and Content Decryption* is established using the methods defined in TS 26.511 [35].

- The selected Adaptation Sets are initialized by downloading the relevant Initialization Segments/CMAF Headers through M4d in the Media Playback Platform as in TS 26.511 [35] establishing a track buffer for each selected media type.

- Depending on the MPD information and/or M7d configuration, one or more of the following functions may be established:

- Metrics Logging and Collection

- Event Processing and Notification

- Client Metadata handling

- The Media Player is left in the READY state.

An application may use this method to load an MPD and in order to prepare playback. In case of errors notifications, it is up to the application to initiate appropriate actions.

#### 13.2.3.4 Pre-load

This clause defines the preload() method.

The following pre-conditions apply:

- The Media Player is in INITIALIZED or READY state.

An 5GMSd-Aware Application calls preload() to cause the player to begin streaming the media as set by the attach() method in preparation for playing.

The parameters of the method are defined in table 13.2.3.4-1.

Table 13.2.3.4-1: Parameters of preload() method

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| sessionId | string | The media delivery session identifier (as specified in clause 7.3.2 of TS 26.510 [56]) of an initialised downlink media streaming session. |
| urlOrMPD | string | Object | A URL to a valid MPD or a valid MPD as defined in ISO/IEC 23009-1 [32] or TS 26.247 [4].The URL may be augmented by MPD Anchors as defined in clause C.4 of ISO/IEC 23009‑1 [32]. |

The following Media Player Actions are expected:

- If in INITIALIZED state, the attach() method is invoked.

- Depending on the type of the MPD, possibly present anchors as well as the wall-clock time, and other MPD information, the earliest media time span for pre-loading is identified.

- The Access Client schedules and generates requests for the relevant media segments based on the ABR Controller information, as well as the throughput estimation and downloads this media.

- The Segments are downloaded from the corresponding URLs through reference point M4d earliest at the segment availability start time of the Segments.

- The Segments ate appropriately appended to the track buffers as established according to *Media Playback Platform and Content Decryption* APIs, following the description in TS 26.511 [35] for playback requirements.

- Configuration and service description parameters are taking into account, for example the content is continuously loaded to remain at the live edge following the latency requirements provided in the service description setting. Content not at the live edge is removed. For static services, the content is loaded from the beginning up to a suitable buffer duration, possibly as configured, and then downloading is stopped.

- Appropriate notifications and error messages are generated. For details refer to clause 13.2.5.

- Appropriate Status Information is generated. For details refer to clause 13.2.6.

- The Media Player is in PRELOADED state.

An application may use this method to preload media into the player in order minimize the start-up time.

#### 13.2.3.5 Play

This clause defines the play() method.

The following pre-conditions apply:

- The Media Player is in INITIALIZED or READY or PRELOADED or PAUSED state.

An 5GMSd-Aware Application calls play() to cause the player to begin playback of the media as set by the attach() method.

The parameters of the method are defined in table 13.2.3.5-1.

Table 13.2.3.5-1: Parameters of play() method

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| sessionId | string | The media delivery session identifier (as specified in clause 7.3.2 of TS 26.510 [56]) of an initialised downlink media streaming session. |
| urlOrMPD | string | Object | A URL to a valid MPD or a valid MPD as defined in ISO/IEC 23009‑1 [32] or TS 26.247 [4].The URL may be augmented by MPD Anchors as defined in clause C.4 of ISO/IEC 23009-1 [32]. |

The following Media Player Actions are expected:

- If in INITIALIZED state, the attach() method is invoked.

- If in PAUSED state, the earliest media time is MEDIA\_TIME (for details see clause 13.2.3.6), else, depending on the type of the MPD, possibly present anchors as well as the wall-clock time, and other MPD information, the earliest media time for start-up is identified.

- The Access Client checks the available buffer state of media in the Media Playback Platform. Based on this, the Access Client schedules and generates requests for the relevant media segments based on the ABR Controller information, as well as the throughput estimation and downloads this media.

- The Segments are downloaded from the corresponding URLs through M4d earliest at the segment availability start times.

- The media is appropriately appended to the *Media Playback Platform and Content Decryption* APIs, following the description in TS 26.511 [35] for playback requirements.

- Once a threshold for sufficient buffering is reached, the Media Playback platform is initiated to be started, i.e. a playback is initiated, following the description in TS 26.511 [35] for playback requirements.

- The content is continuously streamed, downloaded and played back.

- Appropriate notifications and error messages are generated. For details refer to clause 13.2.5.

- Appropriate Status Information is generated. For details refer to clause 13.2.6.

- The Media Player is in PLAYING state.

An application may use this method to initiate playback of media.

#### 13.2.3.6 Pause

This clause defines pause() method.

The following pre-conditions apply:

- The Media Player is in PLAYING state.

An 5GMSd-Aware Application calls pause() to cause the Media Playback Platform to pause playback.

The parameters of the method are defined in table 13.2.3.6-1.

Table 13.2.3.6-1: Parameters of pause() method

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| sessionId | string | The media delivery session identifier (as specified in clause 7.3.2 of TS 26.510 [56]) of an initialised downlink media streaming session. |

The following Media Player Actions are expected:

- The playback on the playback platform is paused and the media time is maintained as MEDIA\_TIME.

- The Access Client checks the available buffer state of media in the Media Playback Platform. Based on this, the Access Client schedules and generates requests for the relevant media segments based on the ABR Controller information, as well as the throughput estimation and downloads this media.

- The media is downloaded from the corresponding URL through reference point M4d earliest at the segment availability start time of the media.

- The media is appropriately appended to the *Media Playback Platform and Content Decryption* APIs, following the description in TS 26.511 [35] for playback requirements.

- Once the buffers are sufficiently filled, the client stops downloading.

- Appropriate notifications and error messages are generated. For details refer to clause 13.2.5.

- Appropriate Status Information is generated. For details refer to clause 13.2.6.

- The Media Player is in PAUSED state.

An application may use this method to play back media.

#### 13.2.3.7 Seek

This clause defines seek() method.

The following pre-conditions apply:

- The MediaPlayer is in INITIALIZED, READY, PRELOADED or PAUSED state.

An 5GMSd-Aware Application calls seek() to cause the player to go a specific media time.

The parameters of the method are defined in table 13.2.3.7-1.

Table 13.2.3.7-1: Parameters of seek() method

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| sessionId | string | The media delivery session identifier (as specified in clause 7.3.2 of TS 26.510 [56]) of an initialised downlink media streaming session. |
| urlOrMPD | string | Object | A URL to a valid MPD or a valid MPD.The URL may be augmented by MPD Anchors as defined in clause C.4 of ISO/IEC 23009-1 [32]. |
| mediaTime | Unsigned integer | The media time in milliseconds for playback. |

The following Media Player Actions are expected:

- If in INITIALIZED state, the attach() method is carried out.

- If the mediaTime is not accessible return an error ERROR\_MEDIA\_TIME\_NOT\_ACCESSIBLE and terminate the process.

- The earliest media time is set to the mediaTime.

- The state is set to PAUSED.

- The play() command is issued.

An application may use this method to initiate playback of media.

#### 13.2.3.8 Reset

This clause defines the reset() method.

The following pre-conditions apply:

- The Media Player may be in any state.

An 5GMSd-Aware Application calls reset()resets all information related to the media and the Media Presentation described by the MPD is destroyed.

The parameters of the method are defined in table 13.2.3.6-1.

Table 13.2.3.8-1: Parameters of reset() method

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| sessionId | string | The media delivery session identifier (as specified in clause 7.3.2 of TS 26.510 [56]) of an initialised downlink media streaming session. |

The following Media Player Actions are expected:

- The playback on the playback platform terminated.

- All open requests are cancelled.

- All scheduled requests are deleted.

- The current MPD is removed.

- The Media Player is left in the INITIALIZED state.

An application may use this method to terminate the playback of any media.

#### 13.2.3.9 Destroy

This clause defines destroy() method.

The following pre-conditions apply:

- The Media Player may be in any state.

An 5GMSd-Aware Application calls destroy()resets all information related to the media and the network.

The parameters of the method are defined in table 13.2.3.9-1.

Table 13.2.3.8-1: Parameters of destroy() method

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| sessionId | string | The media delivery session identifier (as specified in clause 7.3.2 of TS 26.510 [56]) of an initialised downlink media streaming session. |

The following Media Player Actions are expected:

- The playback on the playback platform terminated.

- All open requests are cancelled.

- All scheduled requests are deleted.

- The current MPD is removed.

- All network information is history is cleared.

- The Media Player is left in the IDLE state.

An application may use this method to terminate the playback of any media clear and download related information.

### 13.2.4 Configurations and settings API

DASH streaming for a particular downlink media delivery session may be configured by the 5GMSd-Aware Application at reference point M7d or by the Media Session Handler at reference point M11d with the parameters provided in table 13.2.4-1. Note that these parameters may be set and they may also be observed.

Table 13.2.4-1: Media Player Configuration API

|  |  |  |
| --- | --- | --- |
| Status | Type | Definition |
| sessionId | string | A media delivery session identifier for the downlink media streaming session that has been initialised using the method specified in clause 13.2.3.2. |
| source | Object | Provides the MPD and all contained information. |
| consumptionMode | Enum | Defines two modes:live: in this case the target latency is maintained, if specified in the service description, according to the parametersvod: in this case the latency is set by the application and the latency settings are ignored. |
| maxBufferTime | Integer | Maximum buffer time in milliseconds for the service.  |
| serviceDescriptionId | id | Selects a service description by selecting an identifier. |
| serviceDescriptions[] | Service description parameters | Configures a service description as defined in annex K of ISO/IEC 23009-1 [32]. This allows the application to define additional service descriptions beyond those defined in the MPD. |
|  | id | id | Sets a service description identifier different from the ones available in the service descriptions in the MPD or modifies existing service descriptions. |
|  | serviceLatency | Object | Sets service description parameters for the service latency, as defined in table K.1 of ISO/IEC 23009-1 [32]. |
|  | playBackRate | Object | Sets service description parameters for the playback rate, as defined in table K.2 of ISO/IEC 23009-1 [32] when the service is consumed in live mode. |
|  | operatingQuality | Object | Sets service description parameters for the operating quality, as defined in table K.3 of ISO/IEC 23009-1 [32]. |
|  | operatingBandwidth | Object | Sets service description parameters for the operating bandwidth, as defined in table K.4 of ISO/IEC 23009-1 [32]. |
| mediaSettings[] | Media type audio, video, subtitle | Sets the selected Adaptation Set based on the available Adaptation Sets for each media type. |
| metricsConfiguration[ ] | Object | Zero or more sets of settings for collecting metrics in relation to the downlink media streaming session. |

### 13.2.5 Notifications and error events

Table 13.2.5-1 provides a list of notification events that are provided by the Media Player to 5GMSd-Aware Applications at reference point M7d and to the Media Session Handler at reference point M11d. Every notification and error event is disambiguated by a media delivery session identifier.

Table 13.2.5-1: Media Player Notification events

|  |  |  |
| --- | --- | --- |
| Status | Definition | Payload |
| AST\_IN\_FUTURE | Triggered when playback will not start yet as the MPD's availabilityStartTime is in the future. | Media delivery session identifier, Time before playback will start. |
| AVAILABLE\_MEDIA\_CHANGED | The list of available media has changed. | Media delivery session identifier, Media type:- video- audio- subtitle- all |
| BUFFER\_EMPTY | Triggered when the media playback platform's buffer state changes to stalled. | Media delivery session identifier, Media Type |
| BUFFER\_LOADED | Triggered when the media playback platform's buffer state changes to loaded. | Media delivery session identifier, Media Type |
| CAN\_PLAY | Sent when enough data is available that the media can be played. | Media delivery session identifier |
| MANIFEST\_LOADED | Triggered when the manifest load is complete | Media delivery session identifier |
| METRIC\_ADDED | Triggered every time a new metric is added. | Media delivery session identifier |
| METRIC\_CHANGED | The minimum bit rate that the ABR algorithms will choose. Use NaN for no limit. | Media delivery session identifier |
| METRIC\_UPDATED | Set to true if you would like DASH Client to keep downloading fragments in the background when the video element is paused. | Media delivery session identifier |
| METRICS\_CHANGED | Triggered whenever there is a change to the overall metrics. | Media delivery session identifier |
| OPERATION\_POINT\_CHANGED | Triggered whenever there is a change of an operation point parameter. | Media delivery session identifier, External reference identifier of currently selected Service Operation Point. |
| PLAYBACK\_ENDED | Sent when playback completes. | Media delivery session identifier |
| PLAYBACK\_ERROR | Sent when an error occurs. The element's error attribute contains more information. | Media delivery session identifier, Error attribute. |
| PLAYBACK\_PAUSED | Sent when playback is paused. | Media delivery session identifier |
| PLAYBACK\_PLAYING | Sent when the media begins to play (either for the first time, after having been paused, or after ending and then restarting). | Media delivery session identifier |
| PLAYBACK\_SEEKED | Sent when a seek operation completes. | Media delivery session identifier |
| PLAYBACK\_SEEKING | Sent when a seek operation begins. | Media delivery session identifier |
| PLAYBACK\_STALLED | Sent when the media playback platform reports stalled | Media delivery session identifier |
| PLAYBACK\_STARTED | Sent when playback of the media starts after having been paused; that is, when playback is resumed after a prior pause event. | Media delivery session identifier |
| PLAYBACK\_WAITING | Sent when the media playback has stopped because of a temporary lack of data. | Media delivery session identifier |
| SERVICE\_DESCRIPTION\_SELECTED | sent when the DASH client has selected a service description. | Media delivery session identifier |
| SERVICE\_DESCRIPTION\_CHANGED | Sent when the DASH client has changed a service description. | Media delivery session identifier |
| SERVICE\_DESCRIPTION\_VIOLATED | Provides notification that the service description parameters are currently not met. | Media delivery session identifier, Parameters of service description that are not met. |
| SOURCE\_INITIALIZED | Triggered when the source is set up and ready. | Media delivery session identifier |

Table 13.2.5-2 provides a list of error events.

Table 13.2.5-2: Media Player Error events

|  |  |  |
| --- | --- | --- |
| Status  | Definition | Payload |
| ERROR\_MPD\_NOT\_FOUND | Triggered when the MPD is not found. | Media delivery session identifier |
| ERROR\_MEDIA\_PLAYBACK | Triggered when there is an error from the media playback platform buffer. | Media delivery session identifier |
| ERROR\_MPD\_NOT\_VALID | The provided MPD is not valid according to the XML schema. | Media delivery session identifier, Detailed error information. |
| ERROR\_MEDIA\_TIME\_NOT\_ACCESSIBLE | After a seek operation, the media time is not accessible. | Media delivery session identifier |
| ERROR\_PROFILE\_NOT\_SUPPORTED | The profile of the Media Presentation is not supported. | Media delivery session identifier |

### 13.2.6 Dynamic Status Information

Table 13.2.6-1 provides a list of dynamically changing status information that can be obtained from the Media Player via reference point M7d or M11d. A separate set of Dynamic Status Information is provided for each active downlink media streaming session, indexed by its media delivery session identifier initialised per clause 13.2.3.2.

Table 13.2.6-1: Media Player Dynamic Status information

|  |  |  |  |
| --- | --- | --- | --- |
| Status | Type | Parameter | Definition |
| state | Enumeration |  | An enumerated value from table 13.2.2‑1 indicating the current state of the Media Player. |
| averageThroughput | float | none | Current average throughput computed in the ABR logic in bit/s. |
| bufferLength | float | MediaType"video", "audio" and "subtitle" | Current length of the buffer for a given media type, in seconds. If no type is passed in, then the minimum of video, audio and subtitle buffer length is returned. NaN is returned if an invalid type is requested, the presentation does not contain that type, or if no arguments are passed and the presentation does not include any adaption sets of valid media type. |
| liveLatency | float | none | Current live stream latency in seconds based on the latency measurement. |
| mediaSetting[ ] | MPDAdaptationSet | MediaType"video", "audio" and "subtitle" | Current media settings for each media type based on the CMAF Header and the MPD information based on the selected Adaptation Set for this media type. |
| mediaTime | float | None | Current media playback time from media playback platform. The media time is in seconds and is relative to the start of the playback and provides the media that is actually rendered. |
| playbackRate | float | None | The current rate of playback. For a video that is playing twice as fast as the default playback, the playbackRate value should be 2.00. |
| availableServiceDescriptions[ ] | Array ofservice descriptions |  | Provides the list of available selectable service descriptions with an id to select from. Those are either configured ones or the ones in the MPD. |
| availableMediaOptions[ ] | List of Adaptation Set or Preselection ids | MediaType"video", "audio" "subtitle""all" | Provides the list of available media options that can be selected by the application based on the capability discovery and the subset information. |
| service‌Operation‌Points | array(Service‌Operation‌Point) |  | The set of Service Operation Points declared in the presentation manifest (e.g. DASH MPD) of the current media presentation. |
| operative‌Service‌Operation‌Point | integer |  | A zero-based index into the service‌Operation‌Points array indicating the Service Operation Point currently operative in the playback session.Set to -1 if the array is empty. |
| metrics[ ][ ] | Metrics |  | A data blob of metrics for each configured metrics collecting scheme. |

Table 13.2.6-2 provides a list of configured operation point information that can be obtained from the client. Any change to a parameter below shall be announced with a notification OPERATION\_POINT\_CHANGED as specified in table 13.2.5‑1.

Table 13.2.6-2: Media Player Service Operation Point Information

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Definition |
| ServiceOperationPoint | Object | The currently configured Service Operation Point parameters according to which the DASH client is operating. |
|  | externalIdentifier | String | The external identifier uniquely identifying this Service Operation Point in the presentation manifest (e.g. DASH MPD). |
|  | mode | Enum | The following operation modes are defined:live: The DASH client operates to maintain configured target latencies using playback rate adjustments and possibly resync.vod: The DASH client operates without latency requirements and rebuffering may result in additional latencies |
|  | maxBufferTime | Integer | maximum buffer time in milliseconds for the service. |
|  | switchBufferTime | Integer | buffer time threshold below which the DASH clients attempt to switch Representations. |
|  | latency | Object | Defines the latency parameters used by the DASH client when operating in live mode. |
|  |  | target | Integer | The target latency for the service in milliseconds. |
|  |  | max | Integer | The maximum latency for the service in milliseconds. |
|  |  | min | Integer | The maximum latency for the service in milliseconds. |
|  | playbackRate | MediaTypeaudio, video, all | Defines the playback rate parameters used by the DASH client for catchup mode and deceleration to avoid buffer underruns and maintaining target latencies. |
|  |  | max | Real | The maximum playback rate for the purposes of automatically adjusting playback latency and buffer occupancy during normal playback, where 1.0 is normal playback speed. |
|  |  | min | Real | The minimum playback rate for the purposes of automatically adjusting playback latency and buffer occupancy during normal playback, where 1.0 is normal playback speed. |
|  | bitRate |  | Defines the operating bit rate parameters used by the DASH client used for a specific media type or aggregated. The values are on IP level. |
|  |  | target | Integer | The target bit rate for the service in bit/s that the client is configured to consume. |
|  |  | max | Integer | The maximum bit rate for the service in bit/s that the client is configured to consume. |
|  |  | min | Integer | The minimum bit rate for the service in bit/s that the client is configured to consume. |
|  | playerSpecificParameters |  | Player-specific parameters may be provided, for example about the used algorithm, etc. |

### 13.2.7 Usage of Media Player information by Media Session Handler

The Media Session Handler may use the notifications, errors and status information provided by the Media Player at reference point M11 to execute relevant tasks, such as compiling QoE metrics reports.

End of changes