**3GPP TSG- Meeting #**

**, , -**

|  |
| --- |
| *CR-Form-v12.3* |
| **CHANGE REQUEST** |
|  |
|  |  | **CR** |  | **rev** |  | **Current version:** |  |  |
|  |
| *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 | **X** |

|  |
| --- |
|  |
| ***Title:***  |  |
|  |  |
| ***Source to WG:*** |  |
| ***Source to TSG:*** | S4 |
|  |  |
| ***Work item code:*** |  |  | ***Date:*** |  |
|  |  |  |  |  |
| ***Category:*** |  |  | ***Release:*** |  |
|  | *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-17 (Release 17)Rel-18 (Release 18)Rel-19 (Release 19) Rel-20 (Release 20)* |
|  |  |
| ***Reason for change:*** | Satisfy the objectives of Work Task 2 “Media delivery from multiple service endpoints/locations” as documented in S4-250411. |
|  |  |
| ***Summary of change:*** | New annexes to introduce Coded Multi-Source Media Format (CMMF) for the purposes of enabling media delivery from multiple service locations within the 5GMS System. These annexes include:a. A New annex that specifies CMMF capabilities and profiles supported by 5GMS. This includes specifying the minimum set of CMMF subatoms, code types, capabilities, and profiles; as well a set of CMMF Content Preparation Templates to encode/decode CMMF objects that conform to these profiles.b. A new annex that provides examples of Media Entry Points containing CMMF configuration information required to be communicated to 5GMS Clients and examples of Content Hosting/Publishing Configurations that implement CMMF within the 5GMS network.c. A new annex that specifies how to map CMMF-enabled content delivery onto standardized HTTP adaptive streaming protocols (e.g., DASH, HLS, etc.). This annex specifies methods to map media resource URLs listed in a media description document (e.g., DASH MPD, HLS playlist, etc.) to and from URLs used to request those CMMF-encoded media resources hosted at service locations within the network. It also specifies a JSON schema that can be used to augment an existing, non-CMMF media description document (e.g., DASH MPD, HLS playlist, etc.) with the necessary CMMF configuration information to perform this mapping and enable a media client to stream CMMF-encoded media. It is proposed that this annex is added to TS 26.512 Release 19 until such a time that ETSI TS 103 973 can be updated to include this functionality. |
|  |  |
| ***Consequences if not approved:*** | Objectives of the Work Item not completely satisfied. |
|  |  |
| ***Clauses affected:*** | G (new), H (new), I (new) |
|  |  |
|  | **Y** | **N** |  |  |
| ***Other specs*** | **X** |  |  Other core specifications  | TS 26.510 CR 0016, TS 26.512 CR 0086 |
| ***affected:*** |  | **X** |  Test specifications |   |
| ***(show related CRs)*** |  | **X** |  O&M Specifications |  |
|  |  |
| ***Other comments:*** |  |
|  |  |
| ***This CR's revision history:*** | S4-250952: New CR. |

===== CHANGE =====

# 2 References

…

[67] ETSI TS 103 973: "Coded Multisource Media Format (CMMF) for Content Distribution and Delivery", October 2024.

[68] IETF RFC 5053: "Raptor Forward Error Correction Scheme for Object Delivery".

[69] IETF RFC 3629: "UTF-8, a transformation format of ISO 10646".

===== CHANGE =====

## 3.3 Abbreviations

…

CMMF Coded Multisource Media Format

…

===== CHANGE =====

Annex G (normative):
CMMF capabilities and profiles

# G.1 General

Coded Multi-source Media Format (CMMF) as specified in ETSI TS 103 973 [67] is an extensible container format designed to facilitate the management and interchange of audio-visual media and metadata in one or more coded representations (e.g., encoded with Application Layer Forward Error Correction, linear, network, or channel codes). The coded media representations supported by CMMF enable the efficient use of multi-source, multi-path, and multi-access connectivity for network-delivered applications.

This annex defines the following multi-source object coding using Coded Multi-Source Media Format (CMMF) specified in ETSI TS 103 973 [67] for use within the 5GMS System:

- 5GMS CMMF encoding and decoding capabilities

- 5GMS CMMF profiles

- 5GMS CMMF Content Preparation Templates

# G.2 CMMF encoding and decoding capabilities

## G.2.1 Overview

The following CMMF minimal encoding and decoding capabilities are defined for use within the 5GMS System:

- CMMF bitstream subatom support

- CMMF code types

## G.2.2 CMMF bitstream subatom support

The following CMMF bitstream subatoms, structures, and parameters defined within these structures shall be supported:

- sync()

- bitstream\_header()

- block\_header()

- packet()

All other subatoms and parameters defined in ETSI TS 103 973 [67] may optionally be supported.

## G.2.3 CMMF code types

The following CMMF code types shall be supported:

- xCD-1 as defined in ETSI TS 103 973 [67] (code\_type 0).

- Raptor as defined in IETF RFC 5053 [68] (code\_type 1).

All other CMMF supported code types (see table 40 of ETSI TS 103 973 [67]) may be optionally supported.

Requirements and/or limitations when using CMMF code\_type 0 are specified in ETSI 103 973 [67] and include:

- Maximum content source size is 264 bytes.

- Maximum number of blocks is 232 + 28 − 1.

- Maximum number of block symbols is 232.

- Maximum block symbol size is 232 bytes.

Requirements and/or limitation when using CMMF code\_type 1 are specified in IETF RFC 5053 [68] and include:

- Maximum content source size is 245 bytes.

- Maximum number of blocks is 216.

- Maximum number of source block symbols is 213.

- Maximum number of block symbols (i.e., maximum Encoding Symbol ID) is 216.

- Maximum block symbol size is 216 − 1 bytes.

## G.2.3 Capability discovery

A 5GMSd Client is expected to support capability discovery such that 5GMS-Aware Applications can identify whether a CMMF profile or the CMMF options used within that profile are supported.

A 5GMSd Client should support at least one of the following capability discovery mechanisms for CMMF profiles:

- If a conforming CMMF subatom or structure is provided for playback initialization and the 5GMSd Client does not throw an error response, then the respective CMMF profile is supported with the requirements defined in a specific clause.

# G.3 5GMS CMMF profiles

### G.3.1 Overview

A 5GMS CMMF profile describes a set of capability and CMMF bitstream construction requirements associated to a service scenario. A default profile is defined in case no other profile is claimed to be supported.

### G.3.2 Downlink streaming default profile

#### G.3.2.1 General

This profile defines the required capabilities and configurations for Content Preparation, Content Hosting, and 5GMSd Client functionalities as defined in clause 4.2.1 of TS 26.501 [2] to enable CMMF multi-source delivery for segmented media within the 5GMS System.

The downlink streaming default profile shall have a CMMF profile\_type (see clause 6.1.4.11 of ETSI TS 103 973 [67]) of 3gpp.5gmsd.a where this value is a string encoded using UTF-8 [69].

#### G.3.2.2 CMMF bitstream/object construction

##### G.3.2.2.1 General

A single media resource (e.g., segment) is first prepared for encoding and encapsulation within a CMMF bitstream/object. Each media resource is treated as a single block for the purposes of encoding it using a supported CMMF code type. The media segment is first partitioned into block\_num\_symbols equal-sized source symbols and then encoded using a supported CMMF code type generating at least block\_num\_symbols coded symbols. Each coded symbol is packaged within an individual CMMF packet subatom.

CMMF bitstreams/objects shall be constructed using the CMMF subatoms and structures as specified in ETSI TS 103 973 [67] and as shown in figure G.3.2.2.1-1. Each CMMF bitstream/object shall begin with the CMMF synchronization (sync()) structure, followed by a bitstream header subatom containing the bitstream\_header() structure, one block header subatom containing the block\_header() structure, and n − 2 greater than or equal to block\_num\_symbols packet subatoms each containing the packet() structure.



Figure G.3.2.2.1-1: CMMF bitstream/object construction for the 5GMS downlink streaming default profile 3gpp.5gmsd.a

The CMMF profile 3gpp.5gmsd.a shall be accompanied by the following profile\_description:

Editor’s Note: If necessary, this will be defined at a later date.

##### G.3.2.2.2 CMMF code\_type parameters

The use of various parameters within the construction of a CMMF bitstream/object depends on the type of CMMF code\_type used. Requirements for the construction of the CMMF bitstream/object based on the code\_type in use are defined in table G.3.2.2.2-1 for code\_type 0 and table G.3.2.2.2-2 for code\_type 1.

Table G.3.2.2.2-1: CMMF code\_type 0 property values for CMMF profile 3gpp.5gmsd.a

|  |  |  |  |
| --- | --- | --- | --- |
| CMMF structure | CMMF property name | Value | Bit field encoding |
| bitstream\_header() | code\_type | 0 | u(4) |
| packet\_header() | packet\_mask: Bit 0 | 0 | v(1) |
| packet\_mask: Bit 4 | 1 | v(1) |
| packet\_symbol\_index | Not defined | Not defined |
| coefficient\_vector() | See clause 5.2.17 of ETSI 103 973 [67] | u(block\_num\_symbols) |
| NOTE: The bit field encoding syntax is described in table 10 of ETSI TS 103 973 [67]. |

Table G.3.2.2.2-2: CMMF code\_type 1 property values for CMMF profile 3gpp.5gmsd.a

|  |  |  |  |
| --- | --- | --- | --- |
| CMMF structure | CMMF property name | Value | Bit field encoding |
| bitstream\_header() | code\_type | 1 | u(4) |
| packet\_header() | packet\_mask: Bit 0 | 1 | v(1) |
| packet\_mask: Bit 4 | 0 | v(1) |
| packet\_symbol\_index | *Encoding Symbol ID* as specified in IETF RFC 5053 [68] | u(16) |
| coefficient\_vector() | Not applicable | Not applicable |
| NOTE: The bit field encoding syntax is described in table 10 of ETSI TS 103 973 [67]. |

##### G.3.2.2.3 CMMF subatom() parameters

Requirements for the construction of the subatom() structure as defined in clause 6.1.2 of ETSI TS 103 973 [67] are defined in table G.3.2.2.3-1. CMMF subatom() properties not specified in the below table are either optional or populated during encoding and packaging of the CMMF bitstream/object.

Table G.3.2.2.3-1: CMMF subatom() property value for CMMF profile 3gpp.5gmsd.a

|  |  |  |
| --- | --- | --- |
| CMMF property name | Value | Bit field encoding  |
| b\_bitstream\_id\_present | 1 | b(1) |
| NOTE: The bit field encoding syntax is described in table 10 of ETSI TS 103 973 [67]. |

##### G.3.2.2.4 CMMF sync() parameters

Requirements for the construction of the sync() structure as defined in clause 6.1.3 of ETSI TS 103 973 [67] are defined in table G.3.2.2.4-1. CMMF sync() properties not specified in the below table are either optional or populated during encoding and packaging of the CMMF bitstream/object.

Table G.3.2.2.4-1: CMMF sync() property value for CMMF profile 3gpp.5gmsd.a

|  |  |  |
| --- | --- | --- |
| CMMF property name | Value | Bit field encoding  |
| b\_content\_encode\_uuid | 1 | b(1) |
| NOTE: The bit field encoding syntax is described in table 10 of ETSI TS 103 973 [67]. |

##### G.3.2.2.5 CMMF bitstream\_header() parameters

Requirements for the construction of the bitstream\_header() structure as defined in clause 6.1.4 of ETSI TS 103 973 [67] are defined in table G.3.2.2.5-1. CMMF bitstream\_header() properties not specified in the below table are either optional or populated during encoding and packaging of the CMMF bitstream/object.

Table G.3.2.2.5-1: CMMF bitstream\_header() property value for CMMF profile 3gpp.5gmsd.a

|  |  |  |
| --- | --- | --- |
| CMMF property name | Value | Bit field encoding  |
| content\_source\_size | See clause G.2.3 | u(64) |
| b\_content\_source\_split | 0 | b(1) |
| code\_type | See clause G.3.2.2.2 | u(4) |
| b\_rfc5052 | 0 | b(1) |
| block\_count\_minus1 | 0 | u(8) |
| b\_content\_block\_separate\_sources | 0 | b(1) |
| b\_profile\_information\_present | 1 | b(1) |
| profile\_type\_size | 12 | u(4) |
| profile\_type | 3gpp.5gmsd.a | v(96) |
| profile\_description | 0 | v(32) |
| NOTE: The bit field encoding syntax is described in table 10 of ETSI TS 103 973 [67]. |

##### G.3.2.2.6 CMMF block\_header() parameters

Requirements for the construction of the block\_header() structure as defined in clause 6.1.5 of ETSI TS 103 973 [67] are defined in table G.3.2.2.6-1. CMMF block\_header() properties not specified in the below table are either optional or populated during encoding and packaging of the CMMF bitstream/object.

Table G.3.2.2.6-1: CMMF block\_header() property value for CMMF profile 3gpp.5gmsd.a

|  |  |  |
| --- | --- | --- |
| CMMF property name | Value | Bit field encoding  |
| block\_size | See clause G.2.3 | u(32) |
| block\_num\_symbols | See clause G.2.3 |  |
| block\_symbol\_size | See NOTE 2 | u(32) |
| b\_block\_max\_symbol\_index\_present | 0 | b(1) |
| b\_block\_content\_source\_index\_present | 0 | b(1) |
| b\_block\_composite\_sources | 0 | b(1) |
| b\_addl\_block\_coding\_info\_present | 0 | b(1) |
| block\_mask: Bit 0 | 1 | v(1) |
| block\_mask: Bit 1 | 0 | v(1) |
| b\_sufficient\_symbols\_present | 1 | b(1) |
| NOTE 1: The bit field encoding syntax is described in table 10 of ETSI TS 103 973 [67]. |
| NOTE 2: The block\_symbol\_size shall be set to the size of each symbol in the block (all symbols are of equal size) according to the following formula: block\_symbol\_size = ceil(block\_size / block\_num\_symbols). |

##### G.3.2.2.7 CMMF packet\_header() parameters

Requirements for the construction of the packet\_header() structure as defined in clause 6.1.7 of ETSI TS 103 973 [67] are defined in table G.3.2.2.7-1. CMMF block\_header() properties not specified in the below table are either optional or populated during encoding and packaging of the CMMF bitstream/object.

Table G.3.2.2.7-1: CMMF packet\_header() property value for CMMF profile 3gpp.5gmsd.a

|  |  |  |
| --- | --- | --- |
| CMMF property name | Value | Bit field encoding  |
| b\_systematic\_symbol | 0 | b(1) |
| packet\_mask: Bit 0 | See clause G.3.2.2.2 | v(1) |
| packet\_mask: Bit 2 | 0 | v(1) |
| packet\_mask: Bit 3 | 0 | v(1) |
| packet\_mask: Bit 4 | See clause G.3.2.2.2 | v(1) |
| packet\_symbol\_index | See clause G.3.2.2.2 | u(16) |
| coefficient\_vector() | See clause G.3.2.2.2 | u(block\_num\_symbols) |
| NOTE 1: The bit field encoding syntax is described in table 10 of ETSI TS 103 973 [67]. |

#### G.3.2.3 CMMF bitstream/object deployment within the 5GMS System

A one-to-many mapping between a media resource (e.g., segment) and the CMMF bitstreams/objects created to deliver that resource shall exist where a 5GMSd Client may fully recover the media resource from:

- Any individual CMMF bitstream/object when it is obtained by the 5GMSd Client in its entirety from one service location exposed at reference point M4d, or

- Some combination of CMMF bitstreams/objects when partial bitstreams/objects are obtained from multiple service locations exposed at reference point M4d.

Each individual CMMF bitstream/object containing a media resource shall consist of a unique encoding of the media resource (i.e., representation or version) such that a 5GMSd Client may jointly decode multiple partially received CMMF bitstreams/objects allowing for the recovery of the media resource in its entirety.

A one-to-one mapping shall exist between each service location exposed by the 5GMSd AS at reference point M4d and the CMMF bitstream/object representation/version hosted at that service location.

Configuration information required by the 5GMSd Client to configure and stream media using CMMF shall be communicated within the Media Player Entry.

### G.3.3 Uplink streaming default profile

#### G.3.3.1 General

This profile defines the required capabilities and configurations for Content Preparation, Content Publishing, and 5GMSu Client functionalities as defined in clause 4.3.1 of TS 26.501 [2] to enable CMMF multi-source delivery for segmented media within the 5GMS System.

The uplink streaming default profile shall have a CMMF profile\_type (see clause 6.1.4.11 of ETSI TS 103 973 [67]) of 3gpp.5gmsu.a where this value is a string encoding using UTF-8 [69].

#### G.3.3.2 CMMF bitstream/object construction

##### G.3.3.2.1 General

A single media resource (e.g., segment) is first prepared for encoding and encapsulation within a CMMF bitstream/object. Each media resource is treated as a single block for the purposes of encoding it using a supported CMMF code type. The media segment is first partitioned into block\_num\_symbols equal-sized source symbols and then encoded using a supported CMMF code type generating at least block\_num\_symbols coded symbols. Each coded symbol is packaged within an individual CMMF packet subatom.

CMMF bitstreams/objects shall be constructed using the CMMF subatoms and structures as specified in ETSI TS 103 973 [67] and as shown in figure G.3.3.2.1-1. Each CMMF bitstream/object shall begin with the CMMF synchronization (sync()) structure, followed by a bitstream header subatom containing the bitstream\_header() structure, one block header subatom containing the block\_header() structure, and *n* *−*2 greater than or equal to block\_num\_symbols packet subatoms each containing the packet() structure.



Figure G.3.3.2.1-1: CMMF bitstream/object construction for the 5GMS uplink streaming default profile 3gpp.5gmsu.a

The CMMF profile 3gpp.5gmsu.a shall be accompanied by the following profile\_description:

Editor’s Note: If necessary, this will be defined at a later date.

##### G.3.3.2.2 CMMF code\_type parameters

The use of various parameters within the construction of a CMMF bitstream/object depends on the type of CMMF code\_type used. Requirements for the construction of the CMMF bitstream/object based on the code\_type in use are defined in table G.3.3.2.2-1 for code\_type 0 and table G.3.3.2.2-2 for code\_type 1.

Table G.3.3.2.2-1: CMMF code\_type 0 property values for CMMF profile 3gpp.5gmsu.a

|  |  |  |  |
| --- | --- | --- | --- |
| CMMF structure | CMMF property name | Value | Bit field encoding |
| bitstream\_header() | code\_type | 0 | u(4) |
| packet\_header() | packet\_mask: Bit 0 | 0 | v(1) |
| packet\_mask: Bit 4 | 1 | v(1) |
| packet\_symbol\_index | Not defined | Not defined |
| coefficient\_vector() | See clause 5.2.17 of ETSI TS 103 973 [67] | u(block\_num\_symbols) |
| NOTE: The bit field encoding syntax is described in table 10 of ETSI TS 103 973 [67]. |

Table G.3.3.2.2-2: CMMF code\_type 1 property values for CMMF profile 3gpp.5gmsu.a

|  |  |  |  |
| --- | --- | --- | --- |
| CMMF structure | CMMF property name | Value | Bit field encoding |
| bitstream\_header() | code\_type | 1 | u(4) |
| packet\_header() | packet\_mask: Bit 0 | 1 | v(1) |
| packet\_mask: Bit 4 | 0 | v(1) |
| packet\_symbol\_index | *Encoding Symbol ID* as specified in IETF RFC 5053 [68] | u(16) |
| coefficient\_vector() | Not applicable | Not applicable |
| NOTE: The bit field encoding syntax is described in table 10 of ETSI TS 103 973 [67]. |

##### G.3.3.2.3 CMMF subatom() parameters

Requirements for the construction of the subatom() structure as defined in clause 6.1.2 of ETSI TS 103 973 [67] are defined in table G.3.3.2.3-1. CMMF subatom() properties not specified in the below table are either optional or populated during encoding and packaging of the CMMF bitstream/object.

Table G.3.3.2.3-1: CMMF subatom() property value for CMMF profile 3gpp.5gmsu.a

|  |  |  |
| --- | --- | --- |
| CMMF property name | Value | Bit field encoding  |
| b\_bitstream\_id\_present | 1 | b(1) |
| NOTE: The bit field encoding syntax is described in table 10 of ETSI TS 103 973 [67]. |

##### G.3.2.2.4 CMMF sync() parameters

Requirements for the construction of the sync() structure as defined in clause 6.1.3 of ETSI TS 103 973 [67] are defined in table G.3.3.2.4-1. CMMF sync() properties not specified in the below table are either optional or populated during encoding and packaging of the CMMF bitstream/object.

Table G.3.2.2.4-1: CMMF sync() property value for CMMF profile 3gpp.5gmsu.a

|  |  |  |
| --- | --- | --- |
| CMMF property name | Value | Bit field encoding  |
| b\_content\_encode\_uuid | 1 | b(1) |
| NOTE: The bit field encoding syntax is described in table 10 of ETSI TS 103 973 [67]. |

##### G.3.2.2.5 CMMF bitstream\_header() parameters

Requirements for the construction of the bitstream\_header() structure as defined in clause 6.1.4 of ETSI TS 103 973 [67] are defined in table G.3.3.2.5-1. CMMF bitstream\_header() properties not specified in the below table are either optional or populated during encoding and packaging of the CMMF bitstream/object.

Table G.3.3.2.5-1: CMMF bitstream\_header() property value for CMMF profile 3gpp.5gmsu.a

|  |  |  |
| --- | --- | --- |
| CMMF property name | Value | Bit field encoding  |
| content\_source\_size | See clause G.2.3 | u(64) |
| b\_content\_source\_split | 0 | b(1) |
| code\_type | See clause G.3.3.2.2 | u(4) |
| b\_rfc5052 | 0 | b(1) |
| block\_count\_minus1 | 0 | u(8) |
| b\_content\_block\_separate\_sources | 0 | b(1) |
| b\_profile\_information\_present | 1 | b(1) |
| profile\_type\_size | 12 | u(4) |
| profile\_type | 3gpp.5gmsu.a | v(96) |
| profile\_description | 0 | v(32) |
| NOTE: The bit field encoding syntax is described in table 10 of ETSI TS 103 973 [67]. |

##### G.3.2.2.6 CMMF block\_header() parameters

Requirements for the construction of the block\_header() structure as defined in clause 6.1.5 of ETSI TS 103 973 [67] are defined in table G.3.3.2.6-1. CMMF block\_header() properties not specified in the below table are either optional or populated during encoding and packaging of the CMMF bitstream/object.

Table G.3.3.2.6-1: CMMF block\_header() property value for CMMF profile 3gpp.5gmsu.a

|  |  |  |
| --- | --- | --- |
| CMMF property name | Value | Bit field encoding  |
| block\_size | See clause G.2.3 | u(32) |
| block\_num\_symbols | See NOTE 2 |  |
| block\_symbol\_size | See clause G.2.3 | u(32) |
| b\_block\_max\_symbol\_index\_present | 0 | b(1) |
| b\_block\_content\_source\_index\_present | 0 | b(1) |
| b\_block\_composite\_sources | 0 | b(1) |
| b\_addl\_block\_coding\_info\_present | 0 | b(1) |
| block\_mask: Bit 0 | 1 | v(1) |
| block\_mask: Bit 1 | 0 | v(1) |
| b\_sufficient\_symbols\_present | 1 | b(1) |
| NOTE 1: The bit field encoding syntax is described in table 10 of ETSI TS 103 973 [67]. |
| NOTE 2: The block\_symbol\_size shall be set to the size of each symbol in the block (all symbols are of equal size) according to the following formula: block\_symbol\_size = ceil(block\_size / block\_num\_symbols). |

##### G.3.3.2.7 CMMF packet\_header() parameters

Requirements for the construction of the packet\_header() structure as defined in clause 6.1.7 of ETSI TS 103 973 [67] are defined in table G.3.3.2.7-1. CMMF block\_header() properties not specified in the below table are either optional or populated during encoding and packaging of the CMMF bitstream/object.

Table G.3.3.2.7-1: CMMF packet\_header() property value for CMMF profile 3gpp.5gmsu.a

|  |  |  |
| --- | --- | --- |
| CMMF property name | Value | Bit field encoding  |
| b\_systematic\_symbol | 0 | b(1) |
| packet\_mask: Bit 0 | See clause G.3.3.2.2 | v(1) |
| packet\_mask: Bit 2 | 0 | v(1) |
| packet\_mask: Bit 3 | 0 | v(1) |
| packet\_mask: Bit 4 | See clause G.3.3.2.2 | v(1) |
| packet\_symbol\_index | See clause G.3.3.2.2 | u(16) |
| coefficient\_vector() | See clause G.3.3.2.2 | u(block\_num\_symbols) |
| NOTE 1: The bit field encoding syntax is described in table 10 of ETSI TS 103 973 [67]. |

#### G.3.3.3 CMMF bitstream/object deployment within the 5GMS System

A one-to-many mapping between a media resource (e.g., segment) and the CMMF bitstreams/objects created to deliver that resource shall exist where the 5GMSu AS or 5GMSu Application Provider may fully recover the media resource from:

- Any individual CMMF bitstream/object when it is obtained from a 5GMSd Client in its entirety from one service location exposed at reference point M4u, or

- Some combination of CMMF bitstreams/objects when partial bitstreams/objects are obtained from a 5GMSu Client from multiple service locations exposed at reference point M4u.

Each individual CMMF bitstream/object containing a media resource shall consist of a unique encoding of the media resource (i.e., representation or version) such that a 5GMSu AS or 5GMSu Application Provider may jointly decode multiple partially received CMMF bitstreams/objects allowing for the recovery of the media resource in its entirety.

A one-to-one mapping shall exist between each service location exposed by the 5GMSu AS at reference point M4u and the CMMF bitstream/object representation/version contributed to that service location.

Configuration information required by the 5GMSd Client to configure and stream media using CMMF shall be communicated within the Media Streamer Entry.

# G.4 5GMS CMMF Content Preparation Templates

### G.4.1 Overview

This clause defines CMMF Content Preparation Templates supported by the Content Preparation Templates Provisioning API specified in clause 5.2.5 of TS 26.510 [56] for the encoding and decoding of CMMF objects by the 5GMS AS for the 5GMS CMMF profiles specified in clause G.3.

- In the case of downlink media streaming, depending on the Content Hosting Configuration, the CMMF Encoder specified in clauses 4.2.1 and 4.3.3 of ETSI TS 103 973 [67] may be a single input, single output or a single input, multiple output process. A media resource (e.g., audio segment, video segment, etc.) made available to the 5GMSd AS at reference point M2d or M10d is ingested into the CMMF Encoder where it is encoded and packaged as one or more CMMF objects, and those objects are made available at service locations exposed by the 5GMSd AS at reference point M4d.

- In the case of uplink media streaming, depending on the Content Publishing Configuration, the CMMF Encoder specified in clauses 4.2.1 and 4.3.3 of ETSI TS 103 973 [67] may be a single input, single output or a single input, multiple output process. A media resource (e.g., audio segment, video segment, etc.) contributed to the 5GMSu AS at a reference point M4u service location is ingested into the CMMF Encoder where it is encoded and packaged as one or more CMMF objects, and those objects are made available by the 5GMSu AS at reference points M2u or M10u. Likewise, the CMMF Decoder specified in clause 4.2.3 and 4.3.3 of ETSI TS 103 973 [67] may be a single input, single output or a multiple input, single output process. CMMF objects (e.g., CMMF encoded audio segments, video segments, etc.) contributed to the 5GMSu AS at one or more reference point M4u service locations is ingested into the CMMF Decoder where they are decoded, and the resulting source media resource is made available by the 5GMSu AS at reference points M2u or M10u.

### G.4.2 Downlink streaming default profile encoder Content Preparation Template

The Content Preparation Template for encoding CMMF objects conforming to the downlink streaming default profile with CMMF profile\_type 3gpp.5gmsd.a provides a generic method of provisioning CMMF content preparation that conforms to the CMMF profile described in clause G.3.2. The Content Preparation Template following this format shall be provisioned using the MIME media type specified in clause J.2.2.

Editor’s Note: To be defined at a later date.

### G.4.3 Uplink streaming default profile encoder Content Preparation Template

The Content Preparation Template for encoding CMMF objects conforming to the uplink streaming default profile with CMMF profile\_type 3gpp.5gmsu.a provides a generic method of provisioning CMMF content preparation that conforms to the CMMF profile described in clause G.3.3. The Content Preparation Template following this format shall be provisioned using the MIME media type specified in clause J.2.3.

Editor’s Note: To be defined at a later date.

### G.4.4 Uplink streaming default profile decoder Content Preparation Template

The Content Preparation Template for decoding CMMF objects conforming to the uplink streaming default profile with CMMF profile\_type 3gpp.5gmsu.a provides a generic method of provisioning CMMF content preparation that conforms to the CMMF profile described in clause G.3.3. The Content Preparation Template following this format shall be provisioned using the MIME media type specified in clause J.2.3.

Editor’s Note: To be defined at a later date.

Annex H (informative):
CMMF Content Hosting/Publishing Configuration and Media Entry Point Examples

# H.1 General

Content delivery within the 5GMS System may be augmented using CMMF to enable delivery of media resources through the simultaneous use of multiple service locations exposed by the 5GMS AS at reference point M4. The use of CMMF for content delivery is initiated by the 5GMS Application Provider where the establishment of a Provisioning Session in which a Content Hosting Configuration or Content Publishing Configuration is defined to distribute CMMF objects at reference point M4, in addition to the definition of Media Entry Points containing CMMF configuration information. Media resources may be encoded within CMMF objects by either the 5GMS Application Provider, 5GMS AS, or 5GMS Client depending on the use case. Likewise, the 5GMS Application Provider, 5GMS AS, or 5GMS Client (depending on the use case) may decode received CMMF objects to recover media resources.

This annex provides examples of Media Entry Points, Content Hosting Configurations, and Content Publishing Configurations that support the delivery of CMMF-encoded media within the 5GMS System.

# H.2 Media Entry Point examples

### H.2.1 General

This clause provides several examples showing how CMMF configuration information required by a 5GMS Client may be communicated within a Media Entry Point. When applicable, a general example of a media asset is used, described by the DASH MPD shown in table H.2.1-1, and available at https://example.com/manifest.mpd.

NOTE: Differences between the URLs of media resources and their associated CMMF objects are restricted to differences in the path part of the URL between and including the leading “/” and the final “/” owing to restrictions imposed by the definition of *DistributionConfiguration.PathRewriteRule* specified in the *ContentHostingConfiguration* resource defined in table 8.8.3-1 of TS 26.510 [56].

Table H.2.1-1: Example MPD

|  |
| --- |
| <?xml version="1.0"?><MPD profiles="urn:3GPP:PSS:profile:DASH10" type="static"  minBufferTime="PT10S"  mediaPresentationDuration="PT2H"  availabilityStartTime="2010-04-01T09:30:47Z"  availabilityEndTime="2010-04-07T09:30:47Z"  xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 3GPP-Rel10-MPD.xsd"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns="urn:mpeg:dash:schema:mpd:2011"> <ProgramInformation moreInformationURL="http://www.example.com"> <Title>Example</Title> </ProgramInformation> <Period start="PT0S"> <AdaptationSet mimeType="video/3gpp"> <ContentComponent contentType="video"/> <ContentComponent contentType="audio" lang="en"/> <Representation codecs="s263, samr" bandwidth="256000" id="256"> <BaseURL>"rep1"</BaseURL> <SegmentList duration="1000" timescale="100">  <Initialization sourceURL="seg-init.3gp"/> <SegmentURL media="seg-1.3gp"/> <SegmentURL media="seg-2.3gp"/> <SegmentURL media="seg-3.3gp"/> </SegmentList>  </Representation> <Representation codecs="mp4v.20.9, mp4a.E1" bandwidth="128000" id="128"> <BaseURL>"rep2"</BaseURL> <SegmentList duration="10"> <Initialization sourceURL="seg-init.3gp"/> <SegmentURL media="seg-1.3gp"/> <SegmentURL media="seg-2.3gp"/> <SegmentURL media="seg-3.3gp"/> </SegmentList> </Representation> </AdaptationSet> </Period> <Period start="PT30S"> <SegmentTemplate  duration="10" initialization="seg-init-$RepresentationId$.3gp" media="http://example.com/$RepresentationId$/$Number$.3gp"/> <AdaptationSet mimeType="video/3gpp" codecs="mp4v.20.9, mp4a.E1"> <ContentComponent contentType="video"/> <ContentComponent contentType="audio" lang="en"/> <Representation bandwidth="256000" id="1"/> <Representation bandwidth="128000" id="2"/> </AdaptationSet> </Period></MPD> |

### H.2.2 Extended FDT Schema for CMMF

#### H.2.2.1 Overview

The examples provided in this clause show how a CMMF extended File Delivery Table (EFDT) as specified in clause D.2.3 of ETSI TS 103 973 [67] may be used to explicitly communicate the necessary CMMF configuration information within the Media Entry Point.

#### H.2.2.2 Single file example

The following example shows an EFDT where a single MP4 file may be streamed using CMMF.

Table H.2.2.2-1: Single MP4 CMMF EFDT example

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?><FDTInstance xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="urn:ETSI:CMMF:2023:FDT" xsi:schemaLocation="urn:ETSI:CMMF:2023:FDT extendedFDT.xsd" Expires="2010-04-07T09:30:47Z" Complete="true" ContentType="video/mp4 codecs="mp4v.20.9, mp4a.E1" FEC-OTI-FEC-Encoding-ID="1"> <File Content-Location="https://example.com/video.mp4" TOI="0" Content-Length="64000"> <EncodedObjects type="source" complete="true"> https://example.com/video.mp4 </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-a.com-provider-service.ms.as.3gppservices.org/cmmf-a/video.mp4 </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-b.com-provider-service.ms.as.3gppservices.org/cmmf-b/video.mp4 </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-c.com-provider-service.ms.as.3gppservices.org/cmmf-c/video.mp4 </EncodedObjects> </File></FDTInstance> |

#### H.2.2.3 DASH MPD example

The following example shows an EFDT where the contents of the MPD shown in table H.2.1-1 may be streamed using CMMF.

Table H.2.2.3-1: MPD CMMF EFDT example

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?><FDTInstance xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="urn:ETSI:CMMF:2023:FDT" xsi:schemaLocation="urn:ETSI:CMMF:2023:FDT extendedFDT.xsd" Expires="2010-04-07T09:30:47Z" Complete="true" FEC-OTI-FEC-Encoding-ID="1"> <File Content-Location="https://example.com/rep1/seg-init.3gp" TOI="0"> <EncodedObjects type="source" complete="true"> https://example.com/rep1/seg-init.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-a.com-provider-service.ms.as.3gppservices.org/rep1/cmmf-a /seg-init.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-b.com-provider-service.ms.as.3gppservices.org/rep1/cmmf-b /seg-init.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-c.com-provider-service.ms.as.3gppservices.org/rep1/cmmf-c /seg-init.3gp </EncodedObjects> </File> <File Content-Location="https://example.com/rep1/seg-1.3gp" TOI="1"> <EncodedObjects type="source" complete="true"> https://example.com/rep1/seg-1.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-a.com-provider-service.ms.as.3gppservices.org/rep1/cmmf-a /seg-1.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-b.com-provider-service.ms.as.3gppservices.org/rep1/cmmf-b /seg-1.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-c.com-provider-service.ms.as.3gppservices.org/rep1/cmmf-c /seg-1.3gp </EncodedObjects> </File> <File Content-Location="https://example.com/rep1/seg-2.3gp" TOI="2"> <EncodedObjects type="source" complete="true"> https://example.com/rep1/seg-2.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-a.com-provider-service.ms.as.3gppservices.org/rep1/cmmf-a /seg-2.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-b.com-provider-service.ms.as.3gppservices.org/rep1/cmmf-b /seg-2.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-c.com-provider-service.ms.as.3gppservices.org/rep1/cmmf-c /seg-2.3gp </EncodedObjects> </File> <File Content-Location="https://example.com/rep1/seg-3.3gp" TOI="3"> <EncodedObjects type="source" complete="true"> https://example.com/rep1/seg-3.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-a.com-provider-service.ms.as.3gppservices.org/rep1/cmmf-a /seg-3.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-b.com-provider-service.ms.as.3gppservices.org/rep1/cmmf-b /seg-3.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-c.com-provider-service.ms.as.3gppservices.org/rep1/cmmf-c /seg-3.3gp </EncodedObjects> </File> <File Content-Location="https://example.com/rep2/seg-init.3gp" TOI="0"> <EncodedObjects type="source" complete="true"> https://example.com/rep2/seg-init.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-a.com-provider-service.ms.as.3gppservices.org/rep2/cmmf-a /seg-init.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-b.com-provider-service.ms.as.3gppservices.org/rep2/cmmf-b /seg-init.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-c.com-provider-service.ms.as.3gppservices.org/rep2/cmmf-c /seg-init.3gp </EncodedObjects> </File> <File Content-Location="https://example.com/rep2/seg-1.3gp" TOI="1"> <EncodedObjects type="source" complete="true"> https://example.com/rep2/seg-1.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-a.com-provider-service.ms.as.3gppservices.org/rep2/cmmf-a /seg-1.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-b.com-provider-service.ms.as.3gppservices.org/rep2/cmmf-b /seg-1.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-c.com-provider-service.ms.as.3gppservices.org/rep2/cmmf-c /seg-1.3gp </EncodedObjects> </File> <File Content-Location="https://example.com/rep2/seg-2.3gp" TOI="2"> <EncodedObjects type="source" complete="true"> https://example.com/rep2/seg-2.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-a.com-provider-service.ms.as.3gppservices.org/rep2/cmmf-a /seg-2.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-b.com-provider-service.ms.as.3gppservices.org/rep2/cmmf-b /seg-2.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-c.com-provider-service.ms.as.3gppservices.org/rep2/cmmf-c /seg-2.3gp </EncodedObjects> </File> <File Content-Location="https://example.com/rep2/seg-3.3gp" TOI="3"> <EncodedObjects type="source" complete="true"> https://example.com/rep2/seg-3.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-a.com-provider-service.ms.as.3gppservices.org/rep2/cmmf-a /seg-3.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-b.com-provider-service.ms.as.3gppservices.org/rep2/cmmf-b /seg-3.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-c.com-provider-service.ms.as.3gppservices.org/rep2/cmmf-c /seg-3.3gp </EncodedObjects> </File> <File Content-Location="https://example.com/seg-init-1.3gp" TOI="0"> <EncodedObjects type="source" complete="true"> https://example.com/seg-init-1.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-a.com-provider-service.ms.as.3gppservices.org/cmmf-a /seg-init-1.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-b.com-provider-service.ms.as.3gppservices.org/cmmf-b /seg-init-1.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-c.com-provider-service.ms.as.3gppservices.org/cmmf-c /seg-init-1.3gp </EncodedObjects> </File> <File Content-Location="https://example.com/1/1.3gp" TOI="1"> <EncodedObjects type="source" complete="true"> https://example.com/1/1.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-a.com-provider-service.ms.as.3gppservices.org/1/cmmf-a/1.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-b.com-provider-service.ms.as.3gppservices.org/1/cmmf-b/1.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-c.com-provider-service.ms.as.3gppservices.org/1/cmmf-c/1.3gp </EncodedObjects> </File> <File Content-Location="https://example.com/1/2.3gp" TOI="2"> <EncodedObjects type="source" complete="true"> https://example.com/1/2.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-a.com-provider-service.ms.as.3gppservices.org/1/cmmf-a/2.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-b.com-provider-service.ms.as.3gppservices.org/1/cmmf-b/2.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-c.com-provider-service.ms.as.3gppservices.org/1/cmmf-c/2.3gp </EncodedObjects> </File> <File Content-Location="https://example.com/1/3.3gp" TOI="3"> <EncodedObjects type="source" complete="true"> https://example.com/1/3.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-a.com-provider-service.ms.as.3gppservices.org/1/cmmf-a/3.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-b.com-provider-service.ms.as.3gppservices.org/1/cmmf-b/3.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-c.com-provider-service.ms.as.3gppservices.org/1/cmmf-c/3.3gp </EncodedObjects> </File> <File Content-Location="https://example.com/seg-init-2.3gp" TOI="0"> <EncodedObjects type="source" complete="true"> https://example.com/seg-init-2.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-a.com-provider-service.ms.as.3gppservices.org/cmmf-a /seg-init-2.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-b.com-provider-service.ms.as.3gppservices.org/cmmf-b /seg-init-2.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-c.com-provider-service.ms.as.3gppservices.org/cmmf-c /seg-init-2.3gp </EncodedObjects> </File> <File Content-Location="https://example.com/2/1.3gp" TOI="1"> <EncodedObjects type="source" complete="true"> https://example.com/2/1.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-a.com-provider-service.ms.as.3gppservices.org/2/cmmf-a/1.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-b.com-provider-service.ms.as.3gppservices.org/2/cmmf-b/1.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-c.com-provider-service.ms.as.3gppservices.org/2/cmmf-c/1.3gp </EncodedObjects> </File> <File Content-Location="https://example.com/2/2.3gp" TOI="2"> <EncodedObjects type="source" complete="true"> https://example.com/2/2.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-a.com-provider-service.ms.as.3gppservices.org/2/cmmf-a/2.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-b.com-provider-service.ms.as.3gppservices.org/2/cmmf-b/2.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-c.com-provider-service.ms.as.3gppservices.org/2/cmmf-c/2.3gp </EncodedObjects> </File> <File Content-Location="https://example.com/2/3.3gp" TOI="3"> <EncodedObjects type="source" complete="true"> https://example.com/2/3.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-a.com-provider-service.ms.as.3gppservices.org/2/cmmf-a/3.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-b.com-provider-service.ms.as.3gppservices.org/2/cmmf-b/3.3gp </EncodedObjects> <EncodedObjects type="cmmf" complete="true"> https://distribution-c.com-provider-service.ms.as.3gppservices.org/2/cmmf-c/3.3gp </EncodedObjects> </File></FDTInstance> |

Editor’s Note: Is it possible to condense table H.2.2.2-1 like how a MPD can be condensed? Also, how would one “merge” this with the MPD shown in table H.2.2.1-1 to create a single Media Entry Point?

### H.2.3 HTTP adaptive streaming CMMF Media Entry Point examples

#### H.2.3.1 Overview

The examples provided in this clause show how an HTTP adaptive streaming CMMF configuration schema as specified in clause I.3 can be paired with a URL to a video file, DASH MPD, HLS playlist, etc. to form a Media Entry Point for the purposes of streaming CMMF-encoded media.

#### H.2.3.2 Single file example

The following example shows how the HTTP adaptive streaming CMMF configuration schema can be paired with a URL to a single MP4 file to form a Media Entry Point.

Table H.2.3.2-1: Single file CMMF Entry Point example

|  |
| --- |
| { "mediaResourceInformation": { "mediaResource": "https://example.com/video.mp4\n", "contentType": "text/uri-list", }, "downlinkConfiguration": [ { "serviceLocation": [ { "baseURL": "https://distribution-a.com-provider-service .ms.as.3gppservices.org", "requestPathPattern": "$", "mappedPath": "cmmf-a/",  }, ], "cmmfConfiguration": { "cmmfVersion": 0, "cmmfCodeType": 0, "cmmfProfile": "3gpp.5gmsd.a", }, }, { "serviceLocation": [ { "baseURL": "https://distribution-b.com-provider-service .ms.as.3gppservices.org", "requestPathPattern": "$", "mappedPath": "cmmf-b/",  }, ], "cmmfConfiguration": { "cmmfVersion": 0, "cmmfCodeType": 0, "cmmfProfile": "3gpp.5gmsd.a", }, }, { "serviceLocation": [ { "baseURL": "https://distribution-c.com-provider-service .ms.as.3gppservices.org", "requestPathPattern": "$", "mappedPath": "cmmf-c/",  }, ], "cmmfConfiguration": { "cmmfVersion": 0, "cmmfCodeType": 0, "cmmfProfile": "3gpp.5gmsd.a", }, }, ],} |

#### H.2.3.3 DASH MPD example

The following example shows how the HTTP adaptive streaming CMMF configuration schema can be paired with the MPD shown in table H.2.1-1 to form a Media Entry Point.

Table H.2.3.3-1: DASH MPD CMMF Entry Point example

|  |
| --- |
| { "mediaResourceInformation": { "mediaResource": "https://example.com/manifest.mpd\n", "contentType": "text/uri-list", }, "downlinkConfiguration": [ { "serviceLocation": [ { "baseURL": "https://distribution-a.com-provider-service .ms.as.3gppservices.org", "requestPathPattern": "$", "mappedPath": "cmmf-a/",  }, ], "cmmfConfiguration": { "cmmfVersion": 0, "cmmfCodeType": 0, "cmmfProfile": "3gpp.5gmsd.a", }, }, { "serviceLocation": [ { "baseURL": "https://distribution-b.com-provider-service .ms.as.3gppservices.org", "requestPathPattern": "$", "mappedPath": "cmmf-b/",  }, ], "cmmfConfiguration": { "cmmfVersion": 0, "cmmfCodeType": 0, "cmmfProfile": "3gpp.5gmsd.a", }, }, { "serviceLocation": [ { "baseURL": "https://distribution-c.com-provider-service .ms.as.3gppservices.org", "requestPathPattern": "$", "mappedPath": "cmmf-c/",  }, ], "cmmfConfiguration": { "cmmfVersion": 0, "cmmfCodeType": 0, "cmmfProfile": "3gpp.5gmsd.a", }, }, ],} |

# H.3 Provisioning Session and Content Hosting Configuration examples

### H.3.1 General

This clause provides implementation examples for configuring the 5GMS System to deliver media resources using CMMF from multiple service locations exposed by the 5GMSd AS at reference point M4d.

### H.3.2 Media delivery from multiple service locations using CMMF example

#### H.3.2.1 Overview

The first implementation example shows how CMMF can be used to enable a basic deployment where a single Provisioning Session is configured by the 5GMSd Application Provider to expose multiple service locations. This example assumes the 5GMSd Application Provider provisions the system in the following manner:

1. A 5GMSd AS is provisioned to ingest media resources at reference point M2d

2. Media resources ingested at reference point M2d are encoded within CMMF objects complying with the downlink streaming profile specified in clause G.3.2 using the CMMF Content Preparation Template specified in clause G.4.2.

3. A Media Player Entry containing the necessary CMMF configuration information (see clause H.2.2 and H.3.2) is provided to the 5GMSd Client from a service location exposed by the 5GMSd AS at reference point M4d.

This implementation example is illustrated in figure H.3.2.1-1.



Media Player Entry, MPD, and CMMF-A

Media Player Entry, MPD, and CMMF-B

Media Player Entry, MPD, and CMMF-C

Figure H.3.2.1-1: Basic deployment example for 5GMSd AS ingest, content preparation, and delivery using CMMF

The remainder of this clause provides an implementation of the 5GMS APIs and protocols to realize the implementation example.

#### H.3.2.2 Provisioning Session provisioning and configuration

To configure the 5GMSd AS, one provisioning session is created by the 5GMSd Application Provider using the create Provisioning Session resource operation specified in clause 5.2.2.3 and the API specified in clause 8.2 of TS 26.510 [56]. Example values of the provisioning session API parameters used are shown in table H.3.2.2-1.

Table H.3.2.2-1: Example ProvisioningSession resource parameters

|  |  |  |
| --- | --- | --- |
| Property name | Property value | Assigned by |
| provisioningSessionId | provisioning.session.a | Media AF |
| provisioningSessionType | MS\_DOWNLINK | Media Application Provider |
| externalServiceId | com.‌provider.‌service |
| appId | dash.downlink.streaming |
| NOTE: Values for provisioningSessionId, externalServiceId, and appId are notional and may not reflect actual values. |

#### H.3.2.3 Content Preparation Templates provisioning and configuration

The Media Application Provider provisions one CMMF encoder Content Preparation Template as specified in clause G.4.2 within the provisioning.session.a Provisioning Session using the create Content Preparation Template resource operation specified in clause 5.2.5.2 and the API defined in clause 8.5 of TS 26.510 [56].

Editor’s Note: While these CMMF encoder Content Preparation Templates have yet to be defined, assume that they have the following functionality:

- Content Preparation is triggered upon the receipt of a pull-based request at a service location exposed at reference point M4d.

- The path of the requested resource is available to the Content Preparation Template associated with the service location that received the request.

- Source content (e.g., original video, audio, etc.) is ingested by the 5GMS AS at reference point M2d and encoded within a CMMF object by the Content Preparation Template. A sub-path URL added by the 5GMSd Client into the M4d request URL is used to determine which CMMF “version” or “stripe” is generated. For the purposes of this example, the Content Preparation Template will generate the CMMF-A “version” or “stripe” if a M4d request URL containing the sub-path cmmf-a is received; it will generate the CMMF-B “version” or “stripe” if the URL contains the sub-path cmmf-b is received; and it will generate the CMMF-C “version” or “stripe” if the URL contains the sub-path cmmf-c is received.

- Upon completion of the CMMF encoding operation, the CMMF object is made available to the 5GMS AS for caching and/or delivery to the requesting 5GMSd Client at the service location where the request was received.

Upon successful provisioning of the Content Preparation Template, the Media AF provides the cmmf.content.preparation.template Content Preparation Template ID to the Media Application Provider.

#### H.3.2.4 Server Certificates provisioning and configuration

The Media Application Provider provisions the Server Certificates for the Provisioning Session using the create Server Certificate resource operation specified in clause 5.2.4.2 and the API defined in clause 8.4 of TS 26.510 [56]. The Server Certificates assume a canonical domain name as specified in clause 6.0.2.2. Example values for the certificates generated by the Media AF are provided in table H.3.2.4-1.

Table H.3.2.4-1: Example Server Certificates parameters

|  |  |  |
| --- | --- | --- |
| CN | subjectAltName | certificateId |
| \*.com-provider-service.ms.as.3gppservices.org | \*.com-provider-service.ms.as.3gppservices.org | server.certificate.a |

#### H.3.2.5 Content Hosting provisioning and configuration

The Media Application Provider provisions the Content Hosting Configuration for each Provisioning Session using the create Content Hosting Configuration resource operation specified in clause 5.2.8.2 and the API defined in clause 8.8 of TS 26.510 [56]. Provisioning the Content Hosting Configuration in each 5GMS AS is performed according to:

- The example base URL of the Media Application Provider’s origin server is https://origin.media-application-provider.com.

- The 5GMSd Application Provider provides the Media Player Entry document URL via reference point M8d, and the 5GMSd Client can access the media resource from a service location exposed by the 5GMSd AS at reference point M4d. Example Media Player Entry documents are provided in clause H.2.2 and H.2.3.

Table H.3.2.5-1 provides example values for the Content Hosting Configuration API parameters.

Table H.3.2.5-1: ContentHostingConfiguration resource parameters

|  |  |  |
| --- | --- | --- |
| Property name | Property value | Assigned by |
| name | content-hosting-configuration-a | Media Application Provider |
| ingestConfiguration |
|  | mode | PULL | Media Application Provider |
|  | protocol | urn:3gpp:‌5gms:‌content-protocol:‌http-pull |
|  | baseURL | https://origin.media-application-provider.com |
| distributionConfiguration |
|  | affinityGroup | affinity.group.a | Media Application Provider |
|  | contentPreparationTemplateId | cmmf.content.preparation.template |
|  | certificateId | server.certificate.a |
|  | canonicalDomainName | distribution-a.com-provider-service.ms.as.3gppservices.org | Media AF |
|  | baseURL | https://distribution-a.com-provider-service.ms.as.3gppservices.org |
|  | pathRewriteRule |
|  |  | requestPathPattern | (cmmf-a/)$ | Media Application Provider |
|  |  | mappedPath | $ |
| distributionConfiguration |
|  | affinityGroup | affinity.group.b | Media Application Provider |
|  | contentPreparationTemplateId | cmmf.content.preparation.template |
|  | certificateId | server.certificate.a |
|  | canonicalDomainName | distribution-b.com-provider-service.ms.as.3gppservices.org | Media AF |
|  | baseURL | https://distribution-b.com-provider-service.ms.as.3gppservices.org |
|  | pathRewriteRule |
|  |  | requestPathPattern | (cmmf-b/)$ | Media Application Provider |
|  |  | mappedPath | $ |
| distributionConfiguration |
|  | affinityGroup | affinity.group.c | Media Application Provider |
|  | contentPreparationTemplateId | cmmf.content.preparation.template |
|  | certificateId | server.certificate.a |
|  | canonicalDomainName | distribution-c.com-provider-service.ms.as.3gppservices.org | Media AF |
|  | baseURL | https://distribution-c.com-provider-service.ms.as.3gppservices.org |
|  | pathRewriteRule |
|  |  | requestPathPattern | (cmmf-c/)$ | Media Application Provider |
|  |  | mappedPath | $ |

#### H.3.2.6 End-to-end procedures for downlink streaming using CMMF

The downlink streaming from multiple service locations and media processing procedures for downlink streaming specified in clauses 5.2.6 and 7.2 of TS 26.501 [2] are followed to provision, prepare, and distribute content for this implementation example.

#### H.3.2.7 End-to-end URL mapping

Table H.3.2.7-1 provides an example of the end-to-end mapping for requests initiated by the Media Player for a subset of the URLs provided in the example MPD shown in table H.2.1-1.

Table H.3.2.7-1: End-to-End URL mapping example

|  |  |  |
| --- | --- | --- |
| MPD URL | M4d Request URLs | M2d Request URL |
| rep1/seg-1.3gp | https://distribution-a.com-provider-service.ms.as.3gppservices.org/rep1/cmmf-a/seg-1.3gp | https://origin.media-application-provider.com/rep1/seg-1.3gp |
| https://distribution-b.com-provider-service.ms.as.3gppservices.org/rep1/cmmf-b/seg-1.3gp |
| https://distribution-c.com-provider-service.ms.as.3gppservices.org/rep1/cmmf-c/seg-1.3gp |
| http://example.com/1/1.3gp | https://distribution-a.com-provider-service.ms.as.3gppservices.org/1/cmmf-a/1.3gp | https://origin.media-application-provider.com/1/1.3gp |
| https://distribution-b.com-provider-service.ms.as.3gppservices.org/1/cmmf-b/1.3gp |
| https://distribution-c.com-provider-service.ms.as.3gppservices.org/1/cmmf-c/1.3gp |

### H.3.3 Media delivery from multiple service locations using CMMF and 5GMSd AS service chaining at reference point M10d example

#### H.3.3.1 Overview

This implementation example shows how CMMF can be used to enable a deployment where 5GMSd AS service chaining at reference point M10d is provisioned. This example assumes the 5GMSd Application Provider provisions the system in the following manner:

1. A 5GMSd AS is provisioned to serve as an origin shield intended to reduce requests for content at reference point M2d. Furthermore, the creation of CMMF objects from ingested content at reference point M2d using Content Preparation is performed within this 5GMSd AS.

2. Two 5GMSd AS’s are provisioned to serve 5GMSd Clients from exposed service locations at reference point M4d. Both 5GMSd AS’s are configured to ingest CMMF encoded content (unique to their exposed service locations) from the 5GMSd AS serving as the origin shield at reference M10d.

2. Media resources ingested at reference point M2d are encoded within CMMF objects complying with the downlink streaming profile specified in clause G.3.2 using the CMMF Content Preparation Template specified in clause G.4.2.

3. A Media Player Entry containing the necessary CMMF configuration information (see clause H.2.2 and H.3.2) is provided to the 5GMSd Client from a service location exposed by the 5GMSd AS at reference point M4d.

This implementation example is illustrated in figure H.3.3.1-1.



CMMF-C

CMMF-B

Media Player Entry and MPD

MPD URL

Figure H.3.3.1-1: Centralized 5GMSd AS content preparation and ingest deployment example

The remainder of this clause provides an implementation of the 5GMS APIs and protocols to realize the implementation example.

#### H.3.3.2 Provisioning Session provisioning and configuration

To configure the 5GMS AS, three provisioning sessions are created by the 5GMSd Application Provider using the create Provisioning Session resource operation specified in clause 5.2.2.3 and the API specified in clause 8.2 of TS 26.510 [56], one for each 5GMS AS. In the interest of space, example values for the three provisioning session API parameters used are shown in a single table, table H.3.3.2-1.

Table H.3.3.2-1: ProvisioningSession resource parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Property name | PS A(Trusted DN) | PS B(Trusted DN) | PS C(External DN) | Assigned by |
| provisioningSessionId | provisioning.session.a | provisioning.session.b | provisioning.session.c | Media AF |
| provisioningSessionType | MS\_DOWNLINK | MS\_DOWNLINK | MS\_DOWNLINK | Media Application Provider |
| externalServiceId | com.‌provider.‌service.a | com.‌provider.‌service.b | com.‌provider.‌service.c |
| appId | dash.downlink.streaming | dash.downlink.streaming | dash.downlink.streaming |
| NOTE: Values for provisioningSessionId, externalServiceId, and appId are notional and may not reflect actual values. |

#### H.3.3.3 Content Preparation Templates provisioning and configuration

The Media Application Provider provisions one CMMF encoder Content Preparation Template as specified in clause G.4.2 within the provisioning.session.a Provisioning Session using the create Content Preparation Template resource operation specified in clause 5.2.5.2 and the API defined in clause 8.5 of TS 26.510 [56].

Editor’s Note: While the CMMF encoder Content Preparation Template has yet to be defined, assume that it has the following functionality:

- Content Preparation is triggered upon the receipt of a pull-based request at reference point M10d.

- The path of the requested resource is available to the Content Preparation Template.

- Source content (e.g., original video, audio, etc.) ingested by the Trusted DN 5GMS AS at reference point M2d is encoded by the Content Preparation Template within a CMMF object. A sub-path URL added by the 5GMSd Client into the M4d request URL is used to determine which CMMF “version” or “stripe” is generated. For the purposes of this example, the Content Preparation Template will generate the CMMF-B “version” or “stripe” if a M4d request URL containing the sub-path cmmf-b is received; and it will generate the CMMF-C “version” or “stripe” if the URL contains the sub-path cmmf-c is received.

- Upon completion of the CMMF encoding operation, the CMMF object is made available to the 5GMS AS for caching and/or delivery to the requesting entity.

Upon successful provisioning of the Content Preparation Template, the Media AF provides the cmmf.content.preparation.template Content Preparation Template ID to the Media Application Provider.

#### H.3.3.4 Server Certificates provisioning and configuration

The Media Application Provider provisions the Server Certificates for each Provisioning Session using the create Server Certificate resource operation specified in clause 5.2.4.2 and the API defined in clause 8.4 of TS 26.510 [56]. The Server Certificates assume a canonical domain name as specified in clause 6.0.2.2. Example values for the certificates generated by the Media AF are provided in table H.3.3.4-1.

Table 4.2.3-1: Server Certificates

|  |  |  |  |
| --- | --- | --- | --- |
| provisioningSessionId | CN | subjectAltName | certificateId |
| provisioning.session.a | \*.com-provider-service-a.ms.as.3gppservices.org | \*.com-provider-service-a.ms.as.3gppservices.org | server.certificate.a |
| provisioning.session.b | \*.com-provider-service-b.ms.as.3gppservices.org | \*.com-provider-service-b.ms.as.3gppservices.org | server.certificate.b |
| provisioning.session.c | \*.com-provider-service-c.ms.as.3gppservices.org | \*.com-provider-service-c.ms.as.3gppservices.org | server.certificate.c |

#### H.3.3.5 Content Hosting provisioning and configuration

The Media Application Provider provisions the Content Hosting Configuration for each Provisioning Session using the create Content Hosting Configuration resource operation specified in clause 5.2.8.2 and the API defined in clause 8.8 of TS 26.510 [56]. Provisioning the Content Hosting Configuration in each 5GMS AS is performed according to:

- The example base URL of the Media Application Provider’s origin server is https://origin.media-application-provider.com.

- The creation of the Content Hosting Configuration for Provisioning Session provisioning.session.a is completed prior to the creation of the Content Hosting Configuration for Provisioning Sessions provisioning.session.b and provisioning.session.c.

- The 5GMSd Application Provider provides the Media Player Entry document URL via reference point M8d, and the 5GMSd Client can access the media resource from a service location exposed by the 5GMSd AS at reference point M4d. Example Media Player Entry documents are provided in clause H.2.2 and H.2.3.

Table H.3.3.5-1 provides example values for the Content Hosting Configuration API parameters for all three Provisioning Sessions.

Table H.3.3.5-1: ContentHostingConfiguration resource parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Property name | PS A | PS B | PS C | Assigned by |
| name | content-hosting-configuration-a | content-hosting-configuration-b | content-hosting-configuration-c | Media Application Provider |
| ingestConfiguration |
|  | mode | PULL | PULL | PULL | Media Application Provider |
|  | protocol | urn:3gpp:‌5gms:‌content-protocol:‌http-pull | urn:3gpp:‌5gms:‌content-protocol:‌http-pull | urn:3gpp:‌5gms:‌content-protocol:‌http-pull |
|  | baseURL | https://origin.media-application-provider.com | https://distribution-a.com-provider-service-a.ms.as.3gppservices.org | https://distribution-a.com-provider-service-a.ms.as.3gppservices.org |
|  | distributionConfiguration |
|  | contentPreparationTemplateId | cmmf.content.preparation.template |  |  | Media Application Provider |
|  | certificateId | server.certificate.a | server.certificate.b | server.certificate.c |
|  | canonicalDomainName | distribution-a.com-provider-service-a.ms.as.3gppservices.org | distribution-a.com-provider-service-b.ms.as.3gppservices.org | distribution-a.com-provider-service-c.ms.as.3gppservices.org | Media AF |
|  | baseURL | https://distribution-a.com-provider-service-a.ms.as.3gppservices.org | https://distribution-a.com-provider-service-b.ms.as.3gppservices.org | https://distribution-a.com-provider-service-c.ms.as.3gppservices.org |
|  | pathRewriteRule |
|  |  | requestPathPattern | (cmmf-[a-c]/)$ |  |  | Media Application Provider |
|  |  | mappedPath | $ |  |  |

#### H.3.3.6 End-to-end procedures for downlink streaming using CMMF

The downlink streaming from multiple service locations and media processing procedures for centralized 5GMSd AS content preparation and ingest specified in clauses 5.2.6 and A.16.3 of TS 26.501 [2] are followed to provision, prepare, and distribute content for this implementation example.

#### H.3.3.7 End-to-end URL mapping

Table H.3.3.7-1 provides an example of the end-to-end mapping for requests initiated by the Media Player for a subset of the URLs provided in the example MPD shown in table H.2.1-1.

Table H.3.3.7-1: End-to-End URL mapping example

|  |  |  |  |
| --- | --- | --- | --- |
| MPD URL | M4d Request URL | M10d Request URL | M2d Request URL |
| rep1/seg-1.3gp | https://distribution-a.com-provider-service-b.ms.as.3gppservices.org/rep1/cmmf-b/seg-1.3gp | https://distribution-a.com-provider-service-b.ms.as.3gppservices.org/rep1/cmmf-b/seg-1.3gp | https://origin.media-application-provider.com/rep1/seg-1.3gp |
| https://distribution-a.com-provider-service-c.ms.as.3gppservices.org/rep1/cmmf-c/seg-1.3gp | https://distribution-a.com-provider-service-c.ms.as.3gppservices.org/rep1/cmmf-c/seg-1.3gp |
| http://example.com/1/1.3gp | https://distribution-a.com-provider-service-b.ms.as.3gppservices.org/1/cmmf-b/1.3gp | https://distribution-a.com-provider-service-b.ms.as.3gppservices.org/1/cmmf-b/1.3gp | https://origin.media-application-provider.com/1/1.3gp |
| https://distribution-a.com-provider-service-c.ms.as.3gppservices.org/1/cmmf-c/1.3gp | https://distribution-a.com-provider-service-c.ms.as.3gppservices.org/1/cmmf-c/1.3gp |

Annex I (normative):
HTTP adaptive streaming using CMMF

# I.1 General

Editor’s Note: This annex is defined in TS 26.512 until such a time that its contents are captured in ETSI TS 103 973 [67].

This annex provides a general method to map CMMF-enabled content delivery onto standardized HTTP adaptive streaming protocols (e.g., DASH, HLS, etc.). This is especially useful for use cases where a 5GMS Application Provider may want to limit modifications to their existing DASH MPDs and/or HLS playlists or avoid creating multiple versions of these documents for every combination of base URLs where those media assets can be accessed (e.g., a 5GMS Application Provider offers many media assets from a large catalogue of content and deploys these assets over multiple networks where they are accessable from different URLs). In particular, this annex specifies how an Access Client maps URLs contained in a generic DASH MPD or HLS playlist to request URLs for CMMF-encoded objects available at reference point M4. It further specifies how 5GMS Client request URLs are used by a CMMF encoder provisioned within the 5GMS AS to produce the necessary CMMF-encoded version or representation of the requested media resource(s).

## I.2 Mapping between presentation manifests and CMMF request URLs

### I.2.1 General

A mapping may be defined that translates URLs of media resources (e.g., Segments) listed within a DASH MDP or HLS playlist into URLs of the CMMF objects encoding those media resources. This clause defines such a mapping for the case when a DASH MPD or HLS playlist references complete media segments, and the case when the DASH MPD or HLS playlist references byte ranges of media segments (e.g., chunked CMAF).

In both cases, a media player’s adaptive bit rate (ABR) algorithm selects an appropriate representation/rendition and time period of the media to download from the contents of the DASH MPD or HLS media playlist. The URL, and possibly the byte range, of the media is identified from the DASH MPD or HLS media playlist, and a mapping is applied to translate the identified media resource URL into a request URL used to access one or more CMMF objects encoding that media resource. These CMMF-encoded objects are requested, downloaded and decoded, and finally the recovered media resource is made available for playback.

### I.2.2 Mapping full object URLs

A media resource (e.g., Segment) and its corresponding URL within the DASH MPD or HLS media playlist is first identified by a media player. The URL defined by the DASH MPD or HLS media playlist is then mapped by the media player into a CMMF object request URL by:

- Adding a scheme and domain name to the URL of the service location for which the CMMF-encoded media resource is hosted or replacing the scheme and domain name of the URL with a scheme and domain name of the service location for which the CMMF-encoded media resource is hosted; and.

- a pre-defined sub-path is inserted into the URL’s path that is associated with the CMMF-encoded media resource. This sub-path may also indicate the CMMF representation or version of the CMMF-encoded media resource to download from the identified service location.

For example, a media resource with the URL https://example.com/path-to-media/segment-1.mp4 defined by a DASH MPD or HLS playlist is mapped to two CMMF-encoded representations or versions of the media resource. The first CMMF-encoded representation or version of the media resource is available at the service location with base URL https://dist-a.com-provider-service.ms.as.3gppservices.org. The sub-path cmmf-a, which also is used to identify which CMMF-encoded representation or version of the media resource to download from the service location, is inserted within the path of the DASH MPD or HLS media playlist URL. Likewise, the second CMMF-encoded representation or version the media resource is available at https://dist-b.com-provider-service.ms.as.3gppservices.org and the sub-path cmmf-b is inserted into the path of the DASH MPD or HLS playlist URL. The mapped request URLs for both CMMF-encoded representations or versions of the media resource are then:

https://dist-a.com-provider-service.ms.as.3gppservices.org/path-to-media/cmmf-a/segment-1.mp4

and

https://dist-b.com-provider-service.ms.as.3gppservices.org/path-to-media/cmmf-b/segment-1.mp4

respectively.

### I.2.3 Mapping sub-object URLs using byte range requests

In the case where a media resource is identified by both a URL to an object and a byte range within that object (e.g., chunked CMAF), the mapping is similar to that described in clause I.2.2. The scheme and domain name to a service location hosting a CMMF-encoded representation or version of the media resource is either added or replaced in the media resource’s DASH MPD or HLS media playlist URL and a sub-path is added to the path. The byte range, which is usually communicated using the Byte-Range HTTP request header, is translated into an additional sub-path element and is also inserted into the mapped URL’s path.

For example, a media resource with the URL https://example.com/path-to-media/segment-1.mp4 and byte range startByte to endByte contained within a DASH MPD or HLS playlist is mapped to two CMMF-encoded representations or versions of the media resource. The first CMMF-encoded representation or version of the media resource is available at the service location with base URL https://dist-a.com-provider-service.ms.as.3gppservices.org. The sub-path cmmf-a, which is also used to identify which CMMF-encoded representation or version of the media resource to download from the service location, is inserted within the path of the DASH MPD or HLS playlist URL. The byte range is also inserted into the path URL as byte-range-*<startByte>*-*<endByte>*. Likewise, the second CMMF-encoded representation or version the media resource is available at https://dist-b.com-provider-service.ms.as.3gppservices.org and the sub-paths cmmf-b and byte-range-*<startByte>*-*<endByte>* are inserted into the path of the DASH MPD or HLS playlist URL. The mapped request URLs for both CMMF-encoded representations or versions of the media resource are:

https://dist-a.com-provider-service.ms.as.3gppservices.org/path-to-media/cmmf-a/byte-range-*<startByte>*-*<endByte>*/segment-1.mp4

and

https://dist-b.com-provider-service.ms.as.3gppservices.org/path-to-media/cmmf-b/byte-range-*<startByte>*-*<endByte>*/segment-1.mp4

respectively.

## I.3 Supplementing presentation manifests with CMMF streaming configuration information

Table I.3-1 specifies a JSON schema that can be used to define the parameters of the mappings between media resources identified within a DASH MPD or HLS media playlist and CMMF-encoded media resources hosted at 5GMS AS service locations. It also allows for different mappings to be specified depending on the media resource type (e.g., video, audio, etc.).

Table I.3-1: HTTP adaptive streaming CMMF configuration information schema

|  |  |
| --- | --- |
| Schema | Description |
| { "mediaResourceInformation": { "mediaResource": string, "contentType": string, }, "downlinkConfiguration": [ { "deliverySystemReference": string, "mediaResourcePathPattern": string, "mediaResourceContentType": string, "serviceLocation": [ { "baseURL": string, "requestPathPattern": string, "mappedPath": string,  }, ], "cmmfConfiguration": { "cmmfVersion": number, "cmmfCodeType": number, "cmmfProfile": string, "cmmfProfileDescription": string, }, }, ],} | REQUIRED. A media resource such as an MPD, HLS media playlist, or URL(s) to a media resource. If this paramater takes the value of "\*\n", the media resource or its URL is provided via a process external to this configuration.REQUIRED. The MIME media type of the mediaResource parameter. For example, the MIME media type for a media Resource with value "\*\n" is "text/plain"; and the MIME media type of a URL, or list of URLs, is "text/uri-list".OPTIONAL. A reference to an entity in an external resource. Used to bind this configuration to the delivery system.OPTIONAL. A regular expression against which the path of the media resource URL shall be compared. If true, this configuration applies. If it is not defined, this configuration applies to all media resource URLs.OPTIONAL. A MIME media type of the media resource for which this downlinkConfiguration applies. If true, this configuration applies. If it is not defined, this configuration applies to all media resource MIME media types.OPTIONAL. Base URL of the service location from which content is made available. Base URLs defined here overwrite the base URLs of the media resource (if they exist) identified above.OPTIONAL. A regular expression against which the path of the media resource URL shall be compared, including the leading “/”, and up to and including the final “/” shall be compared. (Any leaf path element following the final “/” shall be excluded from this comparison.)OPTIONAL. A replacement for the portion of the media resource path that matches the requestPathPattern.REQUIRED if CMMF in use. Version of the CMMF specification in use.OPTIONAL. The CMMF code\_type used (see clause 6.1.4.11 of ETSI TS 103 973 [67]). Note: The cmmfCodeType is provided within the bitstream\_header() subatom of the distributed CMMF object(s).OPTIONAL. The CMMF profile\_type (see clause 6.1.4.11 of ETSI TS 103 973 [67]). Note: the cmmfProfile is provided within the bitstream\_header() subatom of the distributed CMMF object(s).OPTIONAL. The CMMF profile\_description (see clause 6.1.4.12 of ETSI TS 103 973 [67]). Note: The cmmfProfileDescription is provided within the bitstream\_header() subatom of the distributed CMMF object(s). |

NOTE: The above schema may be extended to provide other configuration information. For example, content steering service URLs, protocol types and versions, etc.